Model of client-server information system functioning in the conditions of network reconnaissance

Maximov Roman Viktorovich
Krasnodar Higher Military School
named after the general of the Army
S.M.Shtemenko
Krasnodar, Russia
rvmaxim@yandex.ru

Sokolovsky Sergey Petrovich
Krasnodar Higher Military School
named after the general of the Army
S.M.Shtemenko
Krasnodar, Russia
ssp.vrn@mail.ru

Telenga Alexander Pavlovich
Krasnodar Higher Military School
named after the general of the Army
S.M.Shtemenko
Krasnodar, Russia
telenga@gmail.com

Abstract*— Expansion of possibilities and increase of efficiency of network reconnaissance on opening of client-server information systems actualize questions of maintenance of their stability to influences of destabilizing factors. Known methods of protection against network reconnaissance, based on the implementation of the principles of spatial security, as well as the formalization and implementation of many prohibitive regulations based on the detection and response to the fact of network reconnaissance or computer attacks, are not able to effectively withstand the modern means of network reconnaissance. Implementation of such protection methods forces the attacker to continue influencing the client-server information systems and (or) change the strategy of impact. The article presents a model that allows to study the processes of client-server information system functioning in the conditions of network reconnaissance at various strategies of interacting parties, as well as management of resource opportunities of network reconnaissance facilities when establishing and maintaining network connections. Interaction of conflicting parties is presented in the form of Markovian process with discrete states and continuous time. Elements of novelty of the developed model is the application of the mathematical apparatus of the theory of Markovian processes and the solution of Kolmogorov's equations for the study and solution of the problem of dynamic management of resource opportunities of the client-server information system due to the management of parameters of network connections. The practical importance of the developed model consists in finding the probabilistic and temporal characteristics describing the state of the process of functioning of the client-server information system at various strategies of establishment and maintenance of parameters of connections by the interacting parties.

Keywords—client-server information system; computer attack; network connection; honeypots; network tarpits; protocol; network reconnaissance.

I. Introduction

Currently, a large number of computer attacks are of an intelligence nature used by the attacker with the help of network reconnaissance tools in order to obtain information about the topology and typology of the information system that is the object of the computer attack, as well as the information system security features. Possibilities of network reconnaissance are conditioned by openness of architecture of information systems and protocols of information exchange (TCP/IP family), providing interaction through the organization of interfaces of network reconnaissance with elements of information system. Interfaces allow to carry out interaction of consistently connected devices and programs of the received cumulative system realizing the channel of information leakage [1-6].

The key phases of interaction are software suppression (denial of service), event control (monitoring) and management (control interception). The first phase differs from the third in that it may be declarative in nature, and network reconnaissance may lose the ability to implement dialogue. If the information system is isolated, the means of ensuring the security of information interaction are implemented through computer attacks and undeclared capabilities, ensuring the «delivery» of technical means of network reconnaissance in the infrastructure of the information system (providing contact of technical means of network reconnaissance with the object of protection). Dialogue (software, protocol) interaction is carried out locally from the data link layer of the reference OSI model, and remotely – from the network layer [7-9]. One of the means of network protection, functioning with the use of network strategies aimed at creating illusions of vulnerable targets or contributing to the appearance of more complex (false) infrastructure, are network honeypots. Better ways of misleading include not only providing a plausible target for network reconnaissance, but also such actions as, for example, keeping the connection with the sender of the message packets in two-way order, which causes «exhaustion» of resources of the sender of the message packets to maintain the state of connection, slows down the process of automatic scanning of the information system under attack and, as a result, imposes a restriction on the computing resource used by the offender, which leads to the impossibility of performing the following. The considered methods of protection are implemented in the form of so-called network tarpits [10, 11].

In turn, information security violators are also actively developing and improving tools to reduce the effectiveness of network traps, implementing the following methods of compromising: detecting the unique identifiers (unmasking features) of network traps and detailed analysis of network traffic coming from network traps. Such an unmasking feature of the network «trap» is the use of the value of the TCP-packages «window size» service field, which is set to ten bytes by default [12]. As a means of compromising network «traps» in terms of detecting the fact of using the entire set of IP-addresses, the intruder can use various utilities (nmap, ethereal, arping, etc. specialized software) designed to analyze network traffic and information system topology [13-16].

The purpose of dynamic configuration of information system network connection parameters in the conditions of network reconnaissance is to promptly serve the maximum number of requests of authorized clients with simultaneous reduction of the quality of service of network reconnaissance requests. Since there are no physical level disturbances, it is advisable to determine the disturbing factors of the environment as a set of software interferences and software suppression (computer attacks such as «denial of service», so-called DOS- and DDOS- attacks).

II. Object of study analysis

In order to transfer information between remote information systems, as well as between clients and servers in the information system with client-server architecture, a logical connection is established via communication protocols. The increase in the intensity of incoming information flows from one IP-address, multiple IP-addresses may lead to the implementation of an attack such as «Denial of Service» (DoS), or a distributed attack such as «Distributed Denial of Service» (DDoS), respectively.

Information flow management provides the TCP Internet protocol, it allows you to maintain the reliability of transmission over the TCP protocol by adjusting the speed of information flows between the sender and the recipient of TCP-packages during a particular session. Information flows are controlled by limiting the number of data segments transmitted at one time, as well as requesting confirmation of receipt before sending the next segments.

The connection is initiated by the sender of the message packets. If it is necessary to exchange data with the recipient of message packets, the client application refers to the underlying TCP protocol, which in response to this sends a segment-request to establish a connection to the TCP protocol, working on the side of the sender of the message packets, among other things, the request contains the SYN flag set in «1». After receiving the request, the server allocates certain system resources, setting the initial value of the W_N field «window size» (for example, 25 bytes) for the formation of the TCP header of the response packet of messages, announcing to the sender of the packets of messages about its readiness to receive a certain amount of data, as well as other variables of connection. After all the necessary actions are performed on the server side, the resources are defined, the TCP module sends the client a segment with the flags ACK and SYN with the installed W_N to the sender. In response, the client sends a segment with the ACK flag and switches to the state of the established logical connection.

The duration of the delay may increase and then the server may suspend the information exchange with the client in those periods of time when the information system or server resources (the recipient of TCP-packages of messages) are overloaded. For this purpose the server establishes value of the TCP-buffer by an establishment of a field «window size» in TCP-header of a package of messages equal to zero $W_U = 0$, initiating thereby the mechanism of deduction in the bilateral order of connection with the sender of packages of messages, and directs to it corresponding packages of messages. Initiative reduction of speed of data transmission at each session helps to reduce the conflict of resources of the sender and the recipient of TCP-packages of messages in case of initialization of several sessions of communication. This reduces data loss and the number of redirection of data.

After receiving a message packet with $W_U = 0$, according to the TCP protocol specification, the sender of the message packets will periodically send test single-byte segments, asking the recipient to repeat the information about the size of the window and the expected next byte (the so-called test segment «zero-window probe») to determine when he will be able to resume sending data. By implementing a two-way hold mechanism for the connection with the sender of the message packets, the server may not enlarge the window, leaving it at zero, thereby keeping the sender of the message packets blocked in the long connection for a while until the timeout expires.

If the sender of the message packets wants to break the connection (send a packet of messages with the FIN flag in the TCP header) or send urgent data (send a packet of messages with the URG flag in the TCP header), the recipient of the message packets can ignore these incoming packets of messages by blocking them. The server, while ignoring these packets, forces the client operating system to maintain connection resources until FIN-WAIT-1 expires, waiting for the TCP segment from the server to confirm that it is ready to close the connection. The server (the packet recipient) does not maintain the connection state on its end and does not consume its computing resources, which allows it to fully implement the functions of processing of incoming messages packets from clients with higher priority.

The server sends ACK messages to the client to confirm that the client has received a segment of data (or a set of segments). In the process of information exchange via communication channels it is possible to disrupt the order of segment delivery and loss of segments, which makes it necessary to retransmit them from client to server. In particular, if there is interference in the communication channel, it is possible to send three duplicates of ACK confirmation to each of the received fragments of the TCP-package of messages, which is the use of the saturation control algorithm - the fast repetition algorithm (Fast Retransmit) for the TCP protocol.

Thus, the server strategy is to optimize the distribution of its resources to ensure the timely processing of customer requests, taking into account their priorities, which can be achieved by dynamic management of connection parameters [17].

III. Client-server information system functioning model

Let's assume that there is a node of the information system - a server that ensures the functioning of the client-server system, including in the part of the system of control (evaluation) of the idle time value. The simulated system S changes its state over time (passes from one state to another). Required to study the state of the client-server information system S_1, S_2, … can be listed as follows:

S1 – The client is idle, does not receive and does not transmit message packets;

S2 – Initialization of the connection by the client;

S3 – Estimation of the customer downtime;

S4 – Setting (changing) the data flow rate between the client and the server by setting (changing) the «window size» parameter W;

S5 – Establishing (confirming) the connection by the server and receiving the «window size» parameter W by the client;

S6 – Transmission and reception of data flows between client and server;

S7 – Confirmation by the server of receiving data stream parts (acknowledgement).

Moments of possible transitions of the client-server information system from the state to the state are uncertain, random and occur under the action of the event flows characterized by their intensity λ, presented in Table 1, which are an important characteristic of the event flows and characterize the average number of events per unit of time.

The graph of client-server information system functioning states is shown in Fig. 1.

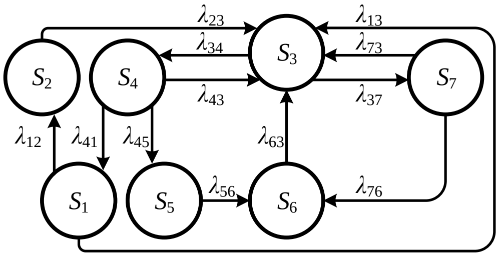

Fig. 1. Graph of client-server information system functioning states

The estimation of efficiency of functioning processes of information system is connected with necessity of modeling of process in real time that causes expediency of use of the mathematical apparatus of Markovian processes which necessary conditions - streams of events are the elementary (possess properties of stationarity, ordinariness and have no consequences). So, the process of client-server information system functioning can be represented as a Markovian process with discrete states and continuous time.

TABLE I. INTENSITY OF EVENT FLOWS IN THE CLIENT-SERVER INFORMATION SYSTEM

λ	Description of the intensity of event flows
λ_{12}	Requests for (new) connection initialization by the client
λ_{23}	Requests for assessment of idle time after the client initializes the connection
λ_{34}	Request for setting (changing) the data flow rate between the client and the server by setting (changing) the «window size» W parameter
λ_{43}	Requests for assessment of idle time after setting (changing) the «window size» W parameter
λ_{45}	Request for the server to establish (confirm) a connection and receive the «window size» W parameter by the client
λ_{56}	Request for transmission and reception of data flows between the client and the server after confirmation of the connection establishment server (W_N and SYN ACK transmission)
λ_{63}	Requests for assessment of idle time in the process of data transmission and reception between the client and the server
λ_{37}	Requests for confirmation by the server of the data stream parts reception (acknowledgement)
λ_{73}	Requests for evaluation of idle time after server confirmation of data stream parts reception (acknowledgement)
λ_{76}	Requests for the client to send the next part of the data stream after the confirmation of the reception of a part of the stream
λ_{41}	Requests for maximization of the client idle time (0-speed data flow parameters) by setting (changing) the «window size» parameter $W = 0$
λ_{13}	Requests for estimation of the value of the client idle time

[a] λ – Intensity of event flows

Let's consider the scenario of transition of the simulated system from the S_i state to the S_j state under the influence of event flows with intensities of λ_{ij}.

When a client-server information system is functioning, objective limitations arise on the performance of both the information system as a whole and its elements. The situation becomes worse when clients with different priorities appear. Therefore they use the dispatching of clients' requests to the server in such information systems. Processing of requests from users with lower priority is suspended without breaking the connection with them if necessary. It is rational, as the repeated establishment of connection causes repetition of the technological operations connected with it that negatively influences on information system productivity.

Let S_1 be the initial state of the simulated client-server information system, in which it does not receive and transmit data flows, i.e. the state of rest, which is characterized for the client by a high value of the idle time, which is evaluated in the state S_3 (on request λ_{13}). It is also reasonable to put the clients to this state of S_1, and the requests received from them are able to overload the server so that they do not have the resource for switching to the state of S_2 by initialization of λ_{12} requests for connection to the server. The clients can initialize alternative requests of λ_{12} until the system resource is exhausted, which will occur if the previous data streams are not closed. If there is still such a resource, the S system switches to the S_2 state and initializes the connection to the server for transmission of message packets with the SYN flag installed. A similar event occurs in the investigated *S* system when new (alternative) authorized clients or new requests from already connected clients appear in it, but via another protocol (organization of a new socket). In this case, λ_{23} requests to estimate the value of S_3 idle time after initialization of the connection by the client appear in the simulated system.

After evaluation of the value of the idle time indicator in the system *S*, λ_{34} requests to set (change) the data flow rate between the client and server S_4 by setting (change) the «window size» parameter W. The value of this parameter is selected in accordance with the value of the idle time indicator of the information system under study: if the connection is initialized by clients with low or usual number of requests, and the server performance has a limited resource, the value of the «window size» parameter W is set to some non-zero value of W_N, for example, $W_N = 20$ bytes. Otherwise, if the connection is initialized by clients with a large number of requests that can overload the server, there is a possibility to set the value of the «window size» parameter W to zero, $W_U = 0$ bytes. As a result, λ_{41} requests for maximizing the idle time value arise in the investigated *S* system. λ_{43} requests for evaluation of the idle time value after establishing (changing) the «window size» parameter W allow to dynamically change (regulate) the data flow rate from clients to the server. If a non-zero value of W_N is set, then λ_{45} requests to establish (confirm) a connection by SYN ACK server and to receive the «window size» parameter W by the clients arise in the investigated system *S*, after which the system switches to the state S_5. This state causes λ_{56} requests to transmit and receive data flows between the clients and the server if the clients (server) have data to be transmitted. As a result, the *S* system under study switches to the state of S_6 of data transmission and reception between clients and the server, which leads to λ_{63} requests for assessment of the idle time value of data transmission and reception between clients and the server. In the process of data streams transmission and reception the clients and the server exchange receipts (confirmations) – the state of S_7 for λ_{37} requests. The acknowledgement procedure also affects the value of the idle time indicator, which is reflected in the column for λ_{73} requests for its evaluation.

After receiving the next λ_{76} receipt from the server, the clients transmit the next part of the data stream to the server. If any parts of the data stream are destroyed in the process of transmission or come from the clients to the server in the wrong order as a result of intentional and/or unintentional interference with the communication channel and the server, the data transmission speed in the client-server information system decreases. In this case, it is not advisable to talk about a *S* system or clients idle, so there is no corresponding link between the states of S_7 and S_1 on the status bar.

A mathematical model of the client-server information system functioning - differential equations with unknown functions $p_i(t)$ - is built on the basis of the marked graph of client-server information system states:

$$\begin{cases} \dfrac{dp_1(t)}{dt} = \lambda_{41} p_4(t) - \lambda_{12} p_1(t) - \lambda_{13} p_1(t), \\ \dfrac{dp_2(t)}{dt} = \lambda_{12} p_1(t) - \lambda_{23} p_2(t), \\ \dfrac{dp_3(t)}{dt} = \lambda_{23} p_2(t) + \lambda_{43} p_4(t) + \lambda_{63} p_6(t) + \\ +\lambda_{73} p_7(t) + \lambda_{13} p_1(t) - (\lambda_{34} + \lambda_{37})\, p_3(t), \\ \dfrac{dp_4(t)}{dt} = \lambda_{34} p_3(t) - (\lambda_{41} + \lambda_{43} + \lambda_{45})\, p_4(t), \\ \dfrac{dp_5(t)}{dt} = \lambda_{45} p_4(t) - \lambda_{56} p_5(t), \\ \dfrac{dp_6(t)}{dt} = \lambda_{56} p_5(t) + \lambda_{76} p_7(t) - \lambda_{63} p_6(t), \\ \dfrac{dp_7(t)}{dt} = \lambda_{37} p_3(t) - (\lambda_{73} + \lambda_{76})\, p_7(t), \\ \sum_{i=1}^{7} p_i(t) = 1. \end{cases} \tag{3}$$

Initial conditions are set to solve Kolmogorov's differential equations. The vector of probabilities of initial states of the Markov chain taking into account absence of influences on the client-server information system at the initial moment of time looks like:

$$p(0) = 1\,0\,0\,0\,0\,0\,0 \tag{4}$$

which corresponds to a high downtime value.

The system of linear differential equations (3) with constant coefficients (homogeneous markov process) is solved by setting numerical values of intensities *λ*, presented in Table 2, and passing to continuous time $t \to \infty$. For any moment *t* the sum of all probabilities of states is equal to one:

$$\sum_{i=1}^{n} p_i(t) = 1 \tag{5}$$

The nature of the selected intensity values is determined in accordance with the strategies of the clients and the server - the parties to the resource conflict.

The client-server information system model takes into account the impact on the server of authorized clients with different request priorities and the number of requests.

The use of the model involves the search for strategies for interaction between the server and the information system clients, and will allow to proceed to the probabilistic assessment of the downtime of the information system and the means of network reconnaissance. Taking into account in the Markov model the time of stay of the information system in each of the states depending on the strategies of the interacting parties allows to study the dynamics of the client-server information system functioning.

The initial data for modeling are the following: the system of linear differential equations (3); the vector of probabilities of initial states (4); the values of the event flows intensity presented in Table 1; the standardization condition (5).

The classical method of the fourth order (the Runge-Kutta method with a fixed integration step which is given to the vector representation) is applied as a method of solving the system of linear differential equations, where each element corresponds to the right side of a certain differential equation in the system.

The use of the known order of solution of the system of linear differential equations by the Runge-Kutta method allows to receive the numerical table of approximate values pi of the sought solutions *p(t)* on some interval $t \in [t_0, t_1]$.

Thus, the probabilistic and temporal characteristics describing the states of the client-server information system functioning process are obtained, which, in their turn, form the basis for the research of this process at various strategies of interacting parties, as shown in Table 3, which allows to evaluate the state of the client-server information system.

TABLE II. EVENT INTENSITY VALUES DEPENDING ON THE CLIENT-SERVER INFORMATION SYSTEM OPERATION STRATEGIES

Features	Strategies			
	C_1	C_2	C_3	C_4
Queue availability, λ_{41}, λ_{45}	**max**	min	**max**	min
Confirmations, λ_{76}	min	min	**max**	**max**

Let's evaluate the model's stability to the variations of the initial data, setting the boundary values in the strategies of the interacting parties, while considering the following variants of strategies:

C_1 is a strategy without confirmation and with a queue, in this case, the connection of clients to the server is carried out via the UDP protocol, where reliable data transfer and confirmation of their receipt, in case of need, must be implemented by the user application, the server receives a significant number of requests for connection from clients and builds them in the queue and then processes them consistently;

C_2 is a strategy without confirmation and without queue, in this case the connection of clients to the server is carried out via the UDP protocol, where reliable data transfer and confirmation of their receipt, in case of need, must be implemented by the user application, the server receives connection requests from clients and manages to process them without delay (without the need to create a queue from the requests) or rejects them;

C_3 is a strategy with the confirmation and with the queue, in this case the connection of clients to the server is carried out via the TCP protocol, where the server receives a significant number of connection requests from clients, builds them in the queue and then sequentially processes them;

C_4 is a strategy with confirmation and without queue, in this case clients with the server is carried out through the TCP protocol, where the server receives connection requests from clients and manages to process them without delay (without the need to create a queue from the requests) or rejects them.

Values of event flows intensity are set as constant according to the chosen strategy of client-server interaction.

The graphs of the probabilities of the client-server information system functioning process states depending on the time $p_1(t)$, $p_2(t)$, …, $p_7(t)$ for the values of event intensities corresponding to the strategy C_1, according to Table 3, are presented in Fig. 2.

TABLE III. NUMERIC TABLE OF APPROXIMATE PROBABILITIES OF BEING IN STATES FOR STRATEGY C_1

n	$[t_0, t_1]$	$p(t)$						
		$p_1(t)$	$p_2(t)$	$p_3(t)$	$p_4(t)$	$p_5(t)$	$p_6(t)$	$p_7(t)$
1	0	1	0	0	0	0	0	0
2	$1 \cdot 10^{-2}$	0,951	$9{,}278 \cdot 10^{-3}$	0,039	$8{,}956 \cdot 10^{-4}$	$2{,}969 \cdot 10^{-5}$	$2{,}041 \cdot 10^{-6}$	$3{,}652 \cdot 10^{-5}$
3	$2 \cdot 10^{-2}$	0,905	0,017	0,074	$3{,}216 \cdot 10^{-3}$	$2{,}148 \cdot 10^{-4}$	$1{,}975 \cdot 10^{-5}$	$1{,}336 \cdot 10^{-4}$
…	…	…	…	…	…	…	…	…
10^3	10	0,513	0,051	0,348	0,053	0,023	$9{,}069 \cdot 10^{-3}$	$2{,}547 \cdot 10^{-3}$

[a.] n – Integration stages

[b.] $[t_0, t_1]$ – Integration Interval Point

At the time interval [0; 0,09] the information system is in the transient mode of functioning, where a spike in the probability values of the state of $p_2(t)$ and $p_3(t)$ is observed, that corresponds to the finding of the information system in the state of initialization of the connection by clients and evaluation of the value of the clients' idle time.

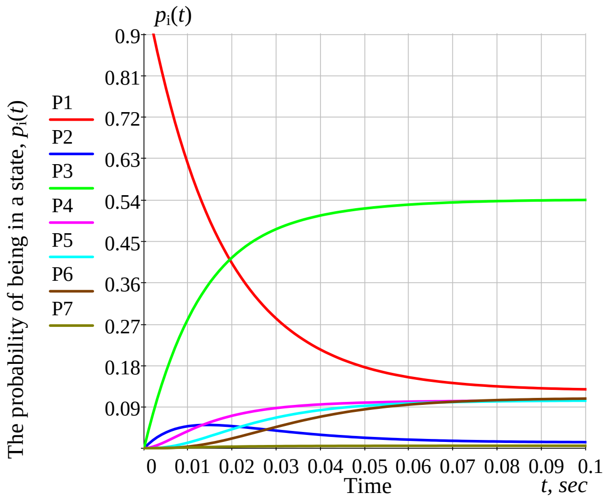

Fig. 2. Results of calculating the dependence of probabilities of states on time for the values of event intensities corresponding to the strategy C_1

In the case of $t \to \infty$, a stationary mode is established in the information system when it randomly changes its states and its probabilities $p_1(t), p_2(t), \ldots, p_7(t)$ no longer depend on time and are equal to the final (limiting) probabilities.

The final probabilities $p_1 = 0{,}513$, $p_2 = 0{,}051$, $p_3 = 0{,}348$, $p_4 = 0{,}053$, $p_5 = 0{,}023$, $p_6 = 9{,}069 \cdot 10^{-3}$, $p_7 = 2{,}547 \cdot 10^{-3}$ show how long the information system stays in different states in average.

In order to study the process of functioning and protection of the information system at the listed strategies of functioning of the client-server information system, and the corresponding values of the intensity of events, the calculation of probabilistic and temporal characteristics is made according to the above example.

We obtain a numerical table of approximate p_i values on the interval $t \in [0, 10]$ with a fixed integration step of $10,^3$ which is presented in Table 4, for the values of event flows of strategy C_2.

TABLE IV. NUMERIC TABLE OF APPROXIMATE PROBABILITIES OF BEING IN STATES FOR STRATEGY C_2

n	**[t_0, t_1]**	***p*(*t*)**						
		$p_1(t)$	$p_2(t)$	$p_3(t)$	$p_4(t)$	$p_5(t)$	$p_6(t)$	$p_7(t)$
1	0	1	0	0	0	0	0	0
2	$1 \cdot 10^{-2}$	0,819	0,086	0,095	$4{,}508 \cdot 10^{-4}$	$1{,}494 \cdot 10^{-5}$	$3{,}407 \cdot 10^{-6}$	$9{,}022 \cdot 10^{-5}$
3	$2 \cdot 10^{-2}$	0,67	0,148	0,179	$1{,}628 \cdot 10^{-3}$	$1{,}085 \cdot 10^{-4}$	$2{,}721 \cdot 10^{-5}$	$3{,}26 \cdot 10^{-4}$
…	…	…	…	…	…	…	…	…
10^3	10	0,061	0,186	0,703	0,027	0,012	$6{,}442 \cdot 10^{-3}$	$5{,}372 \cdot 10^{-3}$

[a.] n – Integration stages

[b.] [t_0, t_1] – Integration Interval Point

Approximate $p_i(t)$ values on the interval $t \in [0, 10]$ with a fixed integration step of 10^3 for the values of intensity of the event flows of the strategy C_3 are presented in numerical Table 5.

TABLE V. NUMERIC TABLE OF APPROXIMATE PROBABILITIES OF BEING IN STATES FOR STRATEGY C_3

n	**[t_0, t_1]**	***p*(*t*)**						
		$p_1(t)$	$p_2(t)$	$p_3(t)$	$p_4(t)$	$p_5(t)$	$p_6(t)$	$p_7(t)$
1	0	1	0	0	0	0	0	0
2	$1 \cdot 10^{-2}$	0,896	$9{,}004 \cdot 10^{-3}$	0,086	$4{,}112 \cdot 10^{-3}$	$1{,}375 \cdot 10^{-4}$	$1{,}458 \cdot 10^{-4}$	$4{,}246 \cdot 10^{-3}$
3	$2 \cdot 10^{-2}$	0,808	0,016	0,15	0,014	$9{,}425 \cdot 10^{-4}$	$1{,}039 \cdot 10^{-3}$	0,014
…	…	…	…	…	…	…	…	…
10^3	10	0,267	0,034	0,316	0,1	0,054	0,092	0,138

[a.] n – Integration stages

[b.] [t_0, t_1] – Integration Interval Point

Approximate $p_i(t)$ values for the values of intensity of the event flows of the strategy C_4 are presented in numerical Table 6.

TABLE VI. NUMERIC TABLE OF APPROXIMATE PROBABILITIES OF BEING IN STATES FOR STRATEGY C_4

n	**[t_0, t_1]**	***p*(*t*)**						
		$p_1(t)$	$p_2(t)$	$p_3(t)$	$p_4(t)$	$p_5(t)$	$p_6(t)$	$p_7(t)$
1	0	1	0	0	0	0	0	0
2	$1 \cdot 10^{-2}$	0,861	0,044	0,091	$8{,}755 \cdot 10^{-5}$	$2{,}913 \cdot 10^{-6}$	$1{,}458 \cdot 10^{-4}$	$4{,}381 \cdot 10^{-3}$
3	$2 \cdot 10^{-2}$	0,741	0,078	0,165	$3{,}07 \cdot 10^{-4}$	$2{,}074 \cdot 10^{-5}$	$1{,}039 \cdot 10^{-3}$	0,015
…	…	…	…	…	…	…	…	…
10^3	10	0,122	0,124	0,465	$3{,}79 \cdot 10^{-3}$	$1{,}811 \cdot 10^{-3}$	0,092	0,191

[a.] n – Integration stages

[b.] [t_0, t_1] – Integration Interval Point

The developed model for the functioning of the client-server information system takes into account the influence and the nature of the influence of the event flows from customers with low and high priority services, as well as the normal and large number of requests from clients to a server that can overload it. The process of protecting the server from overload in accordance with this model is to minimize the likelihood (and average time) of downtime for clients with a high priority of service or a large number of requests, and, therefore, minimize the likelihood of a server overload. Protection of the information system from event flows from clients with a high priority of service or a large number of requests involves the search for strategies for the functioning of the client-server information system depending on the changing options for interaction between the parties due to limited server resources in time. The model makes it possible to identify the dependences of the functioning of the client-server information system on the impact flows, evaluate the effectiveness of the client service, select algorithms for protecting the server from overload, and optimally use the server resources. The increase in the intensity of requests, both from the server and from the clients' side, corresponds to the change of strategies of interacting parties, are presented in Fig. 3.

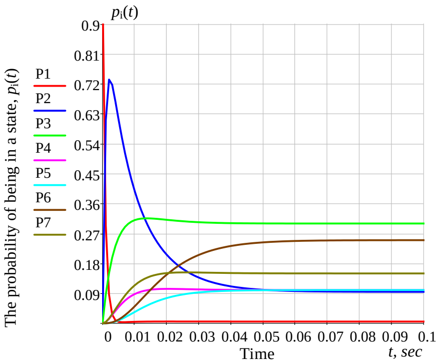

Fig. 3. State probability dependencies on time for given event intensity values (λ_{12}=1751)

With the increase of λ_{12} (to 1751 on Fig. 3) the client-server information system is in a difficult mode of operation, the probability of its being in the state of S_1 is P_1=0,294, and the probability of server overload tends to minimize. The probability of transition of the system to the state S_2 will be maximal and equal to P_2=0,611 for the given values of intensities from the table for strategy C_3. It is possible to reduce the load on this state of the server by increasing its resource by creating a queue of applications between the client and the server by changing the data flow rate by setting the «window size» parameter *W*, regulating the values of the intensity λ_{34} и λ_{41}, as well as by confirming the reception of the data flow parts (acknowledgement) by the server by regulating the values of the intensity λ_{37} и λ_{56}. The diagrams of the probabilities of states versus time for fixed values of event intensities and at λ_{12}=2619 are presented on Fig. 4. If this threshold value (λ_{12}=2619) is exceeded that corresponds to the system state in which the server has received the maximum number of requests, which it can process without overfilling the clipboard or reducing the quality of request servicing, the process of exhausting the system resource due to unfinished connections occurs.

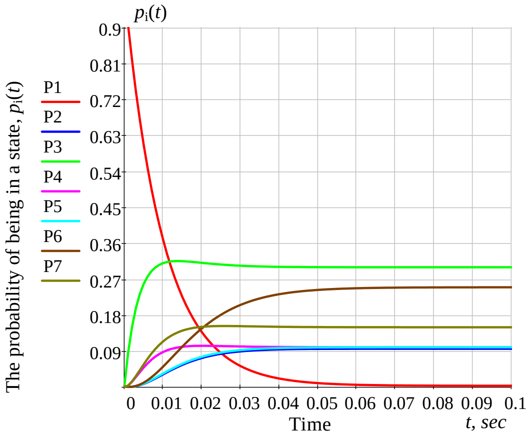

Fig. 4. State probability dependencies on time for given event intensity values (λ_{12}=2619)

IV. Conclusion

The scientific novelty of the model consists in the application of the mathematical apparatus of the Markov random processes theory and the solution of Kolmogorov equations for the study and solution of the problem of dynamic management of the resource potential of the client-server information system due to the management of network connection parameters.

The practical importance lies in finding the probabilistic and temporal characteristics describing the states of the process of functioning of the client-server information system at various strategies of establishment and maintenance of parameters of connections by interacting parties.

References

[1] M. I. Al-Saleh, Z. A. Al-Sharif and L. Alawneh, “Network Reconnaissance Investigation: A Memory Forensics Approach,” 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2019, pp. 36-40. DOI: 10.1109/IACS.2019.8809084.

[2] I. S. Voronchikhin, I. I. Ivanov, R. V. Maximov, S. P. Sokolovsky, “Masking of Distributed Information Systems Structure In Cyber Space,” Voprosy kiberbezopasnosti, 2019, no. 6 (34), pp. 92–101. DOI: 10.21681/2311-3456-2019-6-92-101. (in Russian).

[3] J. H. Jafarian, E. Al-Shaer and Q. Duan, “An Effective Address Mutation Approach for Disrupting Reconnaissance Attacks,” in IEEE Transactions on Information Forensics and Security, vol. 10, no. 12, pp. 2562-2577, Dec. 2015. DOI: 10.1109/TIFS.2015.2467358.

[4] R. Rohrmann, M. W. Patton and H. Chen, “Anonymous port scanning: Performing network reconnaissance through Tor,” 2016 IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, 2016, pp. 217-217. DOI: 10.1109/ISI.2016.7745475.

[5] R. V. Maximov, A. V. Krupenin, S. R. Sharifullin, S. P. Sokolovsky, “Innovative Development of Tools and Technologies to Ensure the Russian Information Security and Core Protective Guidelines,” Voprosy kyberbezopasnosty, 2019, vol. 1, no. 29, pp. 10-17, DOI: 10.21681/2311-3456-2019-1-10-17 (in Russian).

[6] Barabanov A.V., Markov A.S., Tsirlov V.L. Information Security Controls Against Cross-Site Request Forgery Attacks On Software Application of Automated Systems. Journal of Physics: Conference Series. 2018. V. 1015. P. 042034. DOI :10.1088/1742-6596/1015/4/042034

[7] Barabanov A.V., Markov A.S., Tsirlov V.L. Statistics of Software Vulnerability Detection in Certification Testing. Journal of Physics: Conference Series. 2018. V. 1015. P. 042033. DOI :10.1088/1742-6596/1015/4/042033

[8] Q. Hu, M. R. Asghar and N. Brownlee, “Measuring IPv6 DNS Reconnaissance Attacks and Preventing Them Using DNS Guard,” 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg City, 2018, pp. 350-361. DOI: 10.1109/DSN.2018.00045.

[9] S. Achleitner, T. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, R. Chadha, “Cyber Deception: Virtual Networks to Defend Insider Reconnaissance,” 2016 In Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats (MIST '16). ACM, New York, NY, USA, 2016, pp. 57-68. DOI: 10.1145/2995959.2995962.

[10] C. Keil, , M. Nawrocki, T.C. Schmidt, J. Schönfelder, M. Wählisch, “A Survey on Honeypot Software and Data Analysis,” arXiv.org, 2016, vol. 10, pp. 63-75.

[11] P. Sokol, J. Míšek, M. Husák, “Honeypots and honeynets: issues of privacy,” 2017 EURASIP Journal on Information Security, 2017, 1, Article 57 (December 2017), 9 pages. DOI: 10.1186/s13635-017-0057-4.

[12] L. Alt, R. Beverly, A. Dainotti, “Uncovering network tarpits with degreaser,” 2014 In Proceedings of the 30th Annual Computer Security Applications Conference (ACSAC '14). ACM, New York, NY, USA, 2014, pp. 156-165. DOI: 10.1145/2664243.2664285.

[13] S. Laurén, V. Leppänen, S. Rauti, J. Uitto, “A Survey on Anti-honeypot and Anti-introspection Methods,” 2017 Recent Advances in Information Systems and Technologies - Volume 2, WorldCIST'17, Porto Santo Island, Madeira, Portugal, April 11-13, 2017, pp. 125-134. DOI: 10.1007/978-3-319-56538-5_13

[14] B. Nagpal, N. Singh, N. Chauhan and P. Sharma, “CATCH: Comparison and analysis of tools covering honeypots,” 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, 2015, pp. 783-786. DOI: 10.1109/ICACEA.2015.7164809.

[15] M. M. Al-Hakbani and M. H. Dahshan, “Avoiding honeypot detection in peer-to-peer botnets,” 2015 IEEE International Conference on Engineering and Technology (ICETECH), Coimbatore, 2015, pp. 1-7. DOI: 10.1109/ICETECH.2015.7275017.

[16] R. V. Maximov, S. P. Sokolovsky, A. L. Gavrilov, “Hiding computer network proactive security tools unmasking features,” 2017, Selected Papers of the VIII All-Russian Conference with International

Participation “Secure Information Technologies” (BIT 2017), Moscow, Bauman Moscow Technical University Publ., 2017, pp. 88-92.

[17] R. V. Maximov, D. N. Orekhov, S. P. Sokolovsky, “Model and Algorithm of Client-Server Information System Functioning in Network Intelligence Conditions,” Systems of Control, Communication and Security, 2019, no. 4, pp. 50-99. DOI: 10.24411/2410-9916-2019-10403 (in Russian).