SOFTWARE FOR COMPUTER SYSTEMS TROJANS DETECTION AS A SAFETY-CASE TOOL

Sergiy LYSENKO and Oleg SAVENKO

Abstract: This paper presents a behavioural model of Trojans which formalizes the features of Trojans performance in computer systems. The Trojans behavioural model represents its life cycle including three stages: penetration, activation and executing destructive actions. Software for Trojans detection was developed. It is based on methods of detection in 'monitor' and 'scanner' modes. Trojans detection in monitor mode is based on a novel technique for computer system Trojans detection which uses fuzzy logic. It enables a conclusion about the degree of danger of infecting the computer system with Trojans. Trojans detection in a scanner mode is based on a novel technique for constructing the protected sequences and generation of detectors based on algorithms for artificial immune systems. It allows to reveal the fact of system files substitution of Trojans' versions. Trojan detection software allows to detect new Trojans with high degree of reliability and efficiency.

Keywords: Trojans, behavioural model, Trojan life cycle, Trojans detection, fuzzy logic, artificial immune systems, antivirus software.

Introduction

The analysis of the situation of the malware development shows dynamic growth of its quantity. Among them the special place occupies a class of viruses – Trojans which unlike virus programs penetrate into computer system (CS) for the purpose of information plunder that represents real danger.[1] Despite the regular refinement of methods of the search, detecting and removal of Trojans, regular updates of anti-virus bases the numerous facts of the confidential information plunder are observed. The various destructive operations are performed which lead to serious negative consequences. Known methods of the anti-virus software have low efficiency as they are oriented on detection of known Trojans and are insufficiently adapted for recognition of the new suspicious program objects.

That is why the actual problem of safety of various CS is a development of new more perfect safety-case tools which provide increasing of reliability and efficiency of anti-virus software.

INFORMATION & SECURITY. An International Journal, Vol.28, No.1, 2012, 121-132.

Trojan behavioural model

To take into account the features of Trojans and to formalize the process of functioning of Trojans in CS during its life cycle and to take into account the destructive nature of its actions in the CS a new Trojan behavioural model was developed:

$$M_v = \langle \Theta, S, V, L, Aff, \varepsilon, Z \rangle, \quad (1)$$

where Θ - the set of all the Trojans; S - Trojan's life cycle stages (penetration, activation and executing destructive actions), $s_i \in S$, $i = \overline{1,3}$; $V = |V_{mp}|$ - the relationship matrix in which $m = \overline{1,k}$ are functions (mechanisms) which perform ways of the Trojan's penetration to CS via system ports $p = \overline{1,h}$ of network protocols (Table 1); $L = |L_{ab}|$ - a relationship matrix in which $a = \overline{1,\sigma}$ are Trojan's operations which negatively affect the structural components $b = \overline{1,\tau}$ of operating system (Table 2); Aff - a function that defines the interaction between objects of CS and Trojan, thus the set $a \in Aff(b_i, v_j)$ is a set of possible actions, that Trojan causes the object (objects); ε - the ratio between Trojan and its stages, thus if $v \in \Theta$ and $s \in S$ the relationship $v \varepsilon$ S means that Trojan is at the stage S; Z - characteristic parameters of mentioned relations, $Z = \{z_k\}$ - set of destructive actions with normalized priority weights $P = \{p_k\}$ ($\sum p_k = 1$) which take into account the level of actions' danger to the CS. To define the relationship between the Trojans, its actions and stages of its life cycle, the designation was worked in $\longrightarrow$ that means: if $s_i \xrightarrow{a} s_{i+1}$ then the action $a \in A$ causes a transition from s_i stage to stage s_{i+1}. Then the Trojan, which has a life cycle with all stages passes a possible way, is:

$$s_0 \xrightarrow{V,L} s_1 \xrightarrow{V,L} s_2 \xrightarrow{V,L} s_3, \quad (2)$$

where $s_i \xrightarrow{V,L} s_{i+1}$ means the possibility of customized life cycle.[2]

At the stages of activation and executing destructive actions Trojans perform instructions in infected computer system A: $A = \{a_1, a_2, ..., a_{10}\}$, where for example: a_1 – receiving and sending files; a_2 – the creation and deleting files; a_3 – the creation, execution and destruction of system processes; a_4 – the collection and sending system passwords; a_5 – the organization of unauthorized access to Internet;

Table 1: The Matrix of Relationship of Penetration Mechanisms and System Ports V .

p_j \ m_i	m_1	m_2	m_3	m_4	m_5
p_{20}	0	1	0	1	0
p_{25}	1	0	0	0	0
p_{80}	0	0	0	1	0
p_{110}	1	0	0	0	0
p_{119}	0	0	0	0	1

a_6 – reboot, down of the CS; a_7 – the collection and sending of system information (memory size, disk space, operating system version, type of mail client, IP-address, etc.), confidential information (login information for various software); a_8 – downloading other malware; a_9 – recording of user activity and sending reports to intruder; a_{10} – filling the disc space with the unnecessary archived data.

The models of each classes of Trojan with the regard to their specificity and functional significance and are based on its behavioural model were produced. Thus the model of Trojan-Backdoor class is:

$$M_{\Theta_{BD}} = \left\langle \Theta_{BD}, A_{BD}, B_{BD}, W_{\nu}, Inf, X, Y, Z \right\rangle, \tag{3}$$

where Θ_{BD} - set of Trojans of Trojan-Backdoor class; $A_{\Theta_{BD}} = A'_{\Theta_{BD}} \cup A''_{\Theta_{BD}}$ - Trojans' actions, $A'_{\Theta_{BD}}$ - actions of Trojans, which lead to the creation of a new file, $A''_{\Theta_{BD}}$ - actions of Trojans, which lead to the substitution of system files by Trojan versions; $W_{\nu} \in W$ - a set of files sent from the CS by performing actions $a \in A$ which forms a set of signs of incorrect functioning of operating system structural

Table 2: The relationship matrix L of Trojan's actions and the structural components of operating systems

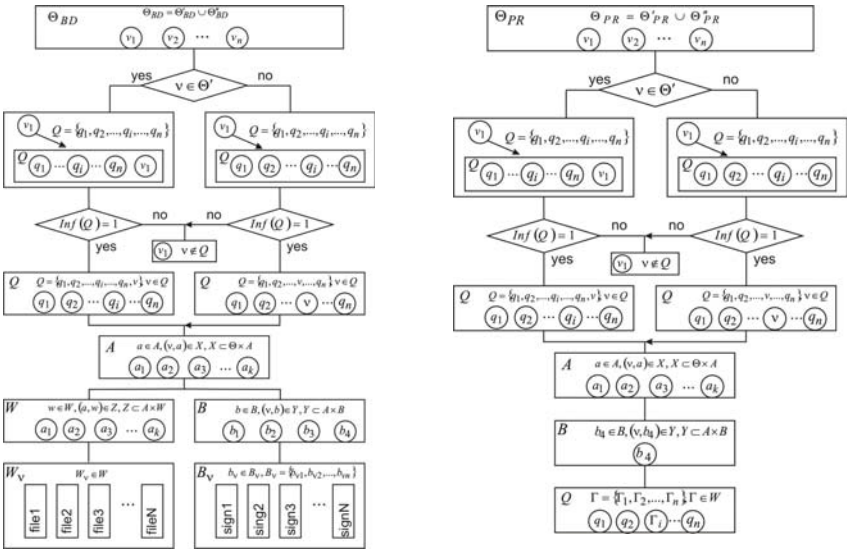
p_j \ m_i	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
b_1	1	1	0	0	0	1	0	0	0	1
b_2	1	0	1	0	0	0	0	0	0	0
b_3	0	1	0	1	0	0	0	0	1	0
b_4	0	0	0	0	1	0	1	1	0	0

components, $b_{BDi} \in B_{BD}$, $B_{BD} = \{b_{BD1}, b_{BD2}, ..., b_{BDn}\}$; Inf - CS sign of infection; $\nu \in \Theta$ - the ratio that describes the implementation of Trojan's actions $a \in A$, $(\nu, a) \in X$, where $X \subset \Theta \times A$; Y - the ratio that describes the implementation of Trojan's actions $a \in A$ and structural components of the OS $(a, b) \in Y$, where $Y \subset A \times B$, Z – ratio between actions $a \in A$ and files $w \in W$, $(a, w) \in Z$, $Z \subset A \times W$.

Trojan-Proxy class carries the setting anonymous access to the Internet. This type of Trojans performs actions $A_{\Theta_{PR}} = \{a_k \mid a_k \in A_{\Theta_{PR}}, k = \overline{1, l}\}$ and opens the Internet access to intruder or access to the infected CS for spamming. Let Γ - set of spam files, $\Gamma = \{\Gamma_1, ..., \Gamma_n\}$, then the model of Trojan-Proxy is:

$$M_{\Theta_{PR}} = \langle \Theta_{PR}, A_{PR}, B_{PR}, \Gamma, Inf, X, Y, Z \rangle. \quad (4).$$

The graph-algorithmic representation of behavioural models of Trojan-Backdoor and Trojan-Proxy are shown in Figure 1.



a) b)

Figure 1: Graf-algorithmic presentation of the behavioural model of Trojan-backdoor (a) and Trojan-proxy (b) classes.

A process of computer systems Trojans detection consists of two sub-processes of monitoring Ω, $\Omega = \{\Omega_1, \Omega_2, \Omega_3, \Omega_4, \Omega_5, \Omega_6\}$ and scanning Δ, $\Delta = \{\Delta_1, \Delta_2, \Delta_3, \Delta_4\}$. The process of Trojan detection in the monitor mode consists of: Ω_1 - monitoring the flows, carried out through the system ports of the CS; Ω_2 - monitoring the execution of the system functions in the CS; Ω_3 - blocking the implementation of application functions, defined as suspicious; Ω_4 - procedure of fuzzification within the fuzzy inference system (FIS) by entering the suspicion degrees and the degrees of computer system infection danger; Ω_5 - implementation of the fuzzy logic engine; Ω_6 - procedure of defuzzification within the FIS to determine the risk of CS infection with Trojans.

The process of Trojan detection in the scanner mode includes the following steps: Δ_1 - forming a files set which must be protected by creating of the set of binary sequences; Δ_2 - generation of a files templates set which is performed by encoding data in a defined format; Δ_3 - generation detectors using one of the artificial immune systems algorithm; Δ_4 - secure scanning of the CS by cooperation the binary sequences with the detectors. In order to formalize the implementation of antivirus detection modes the model of the Trojan it process was developed:

$$M_{\upsilon} = \langle \{E, R, M_W, f_m\}, \{E, H, S, D, E_v, f_s\} \rangle, \quad (5)$$

where for steps $\Omega_1 - \Omega_6$: E - set of diagnosing objects in the monitor mode $e_k \in E$, that is the set of CS files $\Theta \in E$; R - the resultant number $R \in [0,1]$ that indicates the danger degree of infection with Trojan; ε - ratio between objects $\nu \in \Theta$ and stage $s \in S$; M_W - set of behavioural Trojan models; $f_m(I_m, I'_m, I''_m)$ - function of automatically adjusts of computer system detection in monitor mode, whose arguments vary with the input data, where I_m - set of detection information, $I_m = \langle \Theta, V, L, R, \rangle$; I'_m - set of the antivirus diagnostics results, $I'_m = \langle R_1, R_2, ..., R_n \rangle$; I''_m - set of detected malicious software, $I''_m = \langle E, R \rangle$; for steps $\Delta_1 - \Delta_4$: E - set of detection objects in the scanner mode, $e_k \in E$; H - set of objects to be scanned; S - set of protected binary sequences; D - set of generated detectors, $d \in D$, E_v - set of files that were substituted by Trojan versions; $f_s(I_s, I'_s, I''_s)$ - function of automatically adjusts of computer system detection in scanner mode, where I_s - set of detection information, $I_s = \langle H, S, D \rangle$; I'_s - anti-

virus scan results represented with a set of files that were substituted by Trojan versions, $I_s' = \langle E_1, E_2, ..., E_n \rangle$; I_s'' - set of updated system files or installed new software, $I_s'' = \langle E_1', E_2', ..., E_n' \rangle$.[3]

Trojan detection techniques

The scheme of detection process which includes two modes is presented in Figure 2.[4] A new technique for computer system Trojan detection in monitor mode which uses fuzzy logic was developed.[5] It enables to make a conclusion about the danger degree of CS infection with Trojans. For this purpose we construct the input and output linguistic variables with names: "suspicion degree of software" – for the input linguistic variable, and "danger degree of the infection" – for output one.

The task of determination of membership function for input variable we will consider as the task of the ranking for each of mechanisms (functions) m_i of penetration ports p_j with the set of danger indications Z and a choice of the most possible p_j with activation of some function m_i. Then we generate a matrix of advantage $S = |s_{ij}|$. Elements of given matrix s_{ij} are positive numbers: $s_{ij} = s_i / s_j$, $0 < s_{ij} < \infty$; $s_{ji} = 1/s_{ij}$, $s_{ii} = 1$, $i, j = \overline{1,l}$, l - amount of possible results. Elements s_{ij} of matrix S are defined by calculation of values of pair advantages to each indication separately taking into account their scales $Z = \{z_k\}$; $k = \overline{1,r}$ with usage of a formula:

$$s_{ij} = \sum_{k=1}^{r} s_{ij}^{k} \cdot p_k \Big/ \sum_{k=1}^{r} s_{jk}^{k} \cdot p_k . \qquad (6)$$

Eigenvector $\Pi = (\pi_1, ..., \pi_m)$ is defied by using of an advantage matrix. This eigenvector answers maximum positive radical λ of characteristic polynomial $|S - \lambda \cdot E| = 0$. $S \cdot \Pi = \lambda \cdot \Pi$, where E is an identity matrix. Elements of vector Π ($\sum \pi_i = 1$) are identified with an estimation of experts who consider the accepted indications of danger. The same procedure is performed for all m_i. As a result we receive a relationship matrix $V_p = |m_i, p_j|$, in which each pair (relationship) m_i, p_j the value $0 \le \pi \le 1$ responds. Using matrix $V_p = |m_i, p_j|$, we build matrix $V_p^* = |m_i, p_j|$ in which the relationship (m_i, p_j) is used and the elements of this

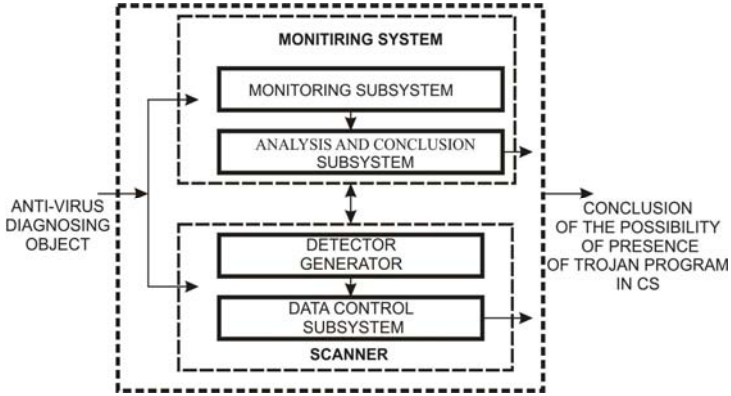


Figure 2: Scheme of the detection process.

relationship have value $\pi_{\max}$ $(0 \le \pi_{\max} \le 1)$. Using matrix $V_p^* = |m_i, p_j|$, we build normalized curve for membership function $\mu_{X^p}(R)$ of the input variable.

Example of possible 20 pairs (x_i, y_j) ranked by the suspicion degree is given in Fig.3. Formation of function membership and at the stages of activation $\mu_{X^a}(R)$ and executing of the destructive actions $\mu_{X^e}(R)$ are similar.

As a part of the solution of the problem the FIS using Mamdani algorithm was realized (Figure 4).

A new technique for constructing the protected sequences and generation of detectors based on the use of algorithms for artificial immune systems was produced.[6] It makes it possible to reveal the fact of system files substitution of Trojans' versions.

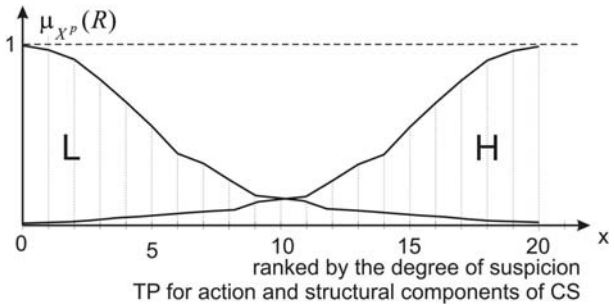


Figure 3: Membership Function of Fuzzy Set "Suspicion Degree of Software."

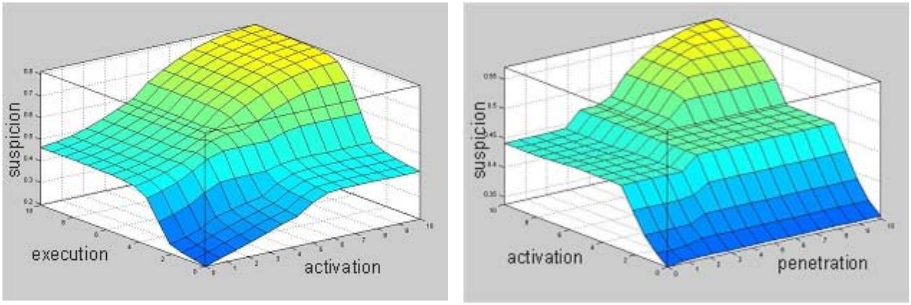


Figure 4: Results of the fuzzy inference system implementation.

The method involves the following steps: forming a set of files to be scanned: system libraries, executables system services and device drivers, which can be taken as the samples; generate protected sequences and detectors depending on operating system; comparison of the protected sequences with detectors at the stage of virus scanning; notification about the substitution when the protected sequences match with detector; check the suspicion of software actions. Thus protected sequences and detectors have format for GNU / Linux operating system:

$$D_i^L = \langle m_1 ... m_i ... m_{x1}, u_1 ... u_i ... u_{x2}, g_1 ... g_i ... g_{x3}, s_1 ... s_i ... s_{x4}, t_1 ... t_i ... t_{x5}, C_1 ... C_i ... C_{x6} \rangle, \quad (7)$$

where $m_1 ... m_i ... m_{x1}$ - file mode (type, permissions); $u_1 ... u_i ... u_{x2}$ - identifier of the file owner; $g_1 ... g_i ... g_{x3}$ - identifier of the group owner; $s_1 ... s_i ... s_{x4}$ - file size; $t_1 ... t_i ... t_{x5}$ - time of last file modification; $C_1 .. C_i ... C_{x6}$ - CRC of the file, $i = \overline{1,n}$, n – number of detectors.

Protected sequences and detectors have format for MS Windows operating system:

$$D_i^W = \langle s_1 ... s_i ... s_{z1}, t_1 ... t_i ... t_{z2}, a_1 ... a_i ... a_{z3}, C_1 ... C_i ... C_{z4} \rangle \quad (8)$$

where $s_1 ... s_i ... s_{z1}$ - file size; $t_1 ... t_i ... t_{z2}$ - time of last file modification; $a_1 ... a_i ... a_{z3}$ - file attribute (read-only, hidden, system, archived); $C_1 .. C_i ... C_{z4}$ - CRC of the file, $i = \overline{1,n}$, n - number of detectors.

Generation of detectors is performed using the modified negative selection algorithm.

The task of determination of Trojan detection efficiency is multi task one. That is why to find the characteristics of structural units of computer system Trojan detection we have to analyze the efficiency based on multiplicative quality criterion. Incorpo-

ration of additional features in the overall detection efficiency is based on the additive criterion. The computer system Trojan detection efficiency is:

$$E = \omega_1 \cdot K_M + \omega_2 \cdot K_S - (\omega_3 \cdot V_M + \omega_4 \cdot V_S + \omega_5 \cdot X_M), \quad (9)$$

where K_M and K_S - efficiency coefficient for the monitor and scanner:

$$K_M = \sqrt[3]{\omega_{11} \cdot D_M \cdot \omega_{12} \cdot T_M \cdot \omega_{13} \cdot T_M^p}, \; K_S = \sqrt[3]{\omega_{21} \cdot D_S \cdot \omega_{22} \cdot T_S \cdot \omega_{23} \cdot T_S^p}, \quad (10)$$

ω - weight coefficient of the additional features and criteria, $\sum \omega_i = 1$; D_M, D_S - reliability of the monitor and scanner; T_M and T_S - detection duration in on-line monitor and scanner mode; T_M^P and T_S^P - time of the preparation for the detection in offline scanner and monitor mode; V_M and V_S - data amount held in the detection process.

$$D_M = \sum_{i=1}^{s} \alpha_i \cdot k_i / \sum_{i=1}^{s} \alpha_i \cdot n_i,$$

where n_i - number of i-th Trojan class, $i = \overline{1,s}$, $s \in N$, k_i - the number of detected Trojans; α_i - percent of i-th Trojan class of all Trojans, $0 \le \alpha_i \le 1$.

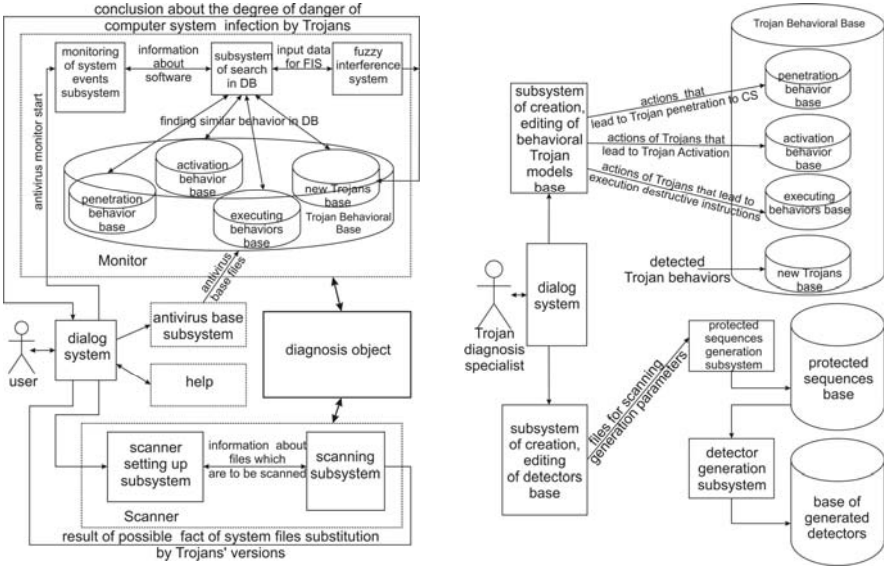
Software and experiments

Antivirus software is based on proposed techniques in monitor and scanner modes with automated setting up of antivirus detection parameters with the improving reliability and efficiency is developed. It allows detecting known and unknown Trojans. The proposed software also allows making a conclusion of regarding the danger degree of CS infection with Trojans and to reveal the fact of system files substitution of Trojans' versions.[7] Functional scheme of software for users and administrators is shown in Figure 5.

Interface results windows of Trojan detection of CS in the monitor and scanner modes are shown in Figure 6.[8] Experimental research of developed software on different processor platforms was conducted. Results demonstrate the possibility of its use in real time on most modern computer systems (Figure 7).

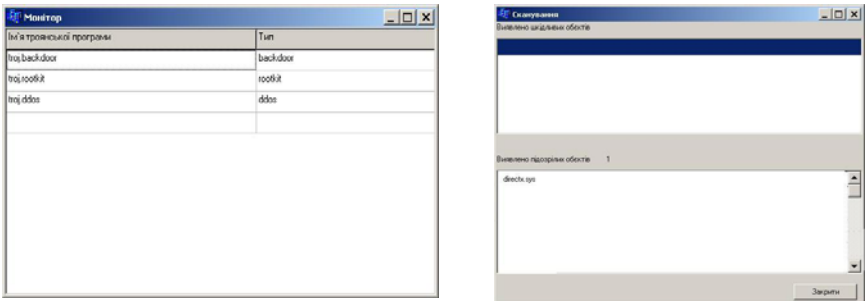
For the experimental determination of the reliability and efficiency 3240 program objects with the Trojans' properties were generated (Table 3).

Reliability of Trojan detection of the developed software (SW) in comparison with known is shown in Figure 8.



a) user mode b) administrator mode

Figure 5: Functional scheme of software for users and administrators.



a) monitor mode b) scanner mode

Figure 6: Interface results windows of Trojan detection in the monitor (a) and scanner (b) modes.

The results confirmed that the use of developed software increases the detection reliability by 5-15%, and efficiency – by 40% in comparison with the known antivirus technologies.

Conclusion

The article is devoted to solving important scientific problem - increasing reliability and efficiency of Trojans detection. A behavioural model of Trojans which formal-

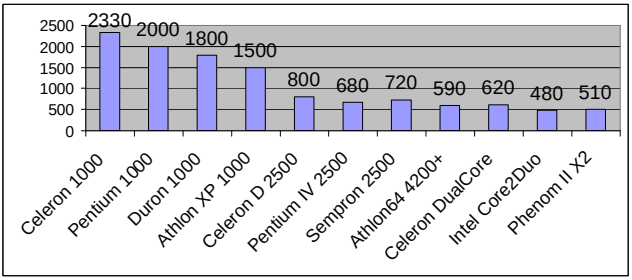


Figure 7: Detection time (in seconds) on different processor platforms.

Table 3. Results of Trojan detection.

Software with Trojan properties	*Detected Trojans*	*Detection rate, %*
Rootkit	180	102	56,67
BackDoor	810	689	85,06
Trojan-PSW	320	251	78,44
Trojan-Clicker	210	139	66,19
Trojan Downloader	850	652	76,71
Trojan-Dropper	230	142	61,74
Trojan-Proxy	150	104	69,33
Trojan-Spy	330	235	71,21
Trojan-Notifier	160	101	63,13
Total	3240	2410	-

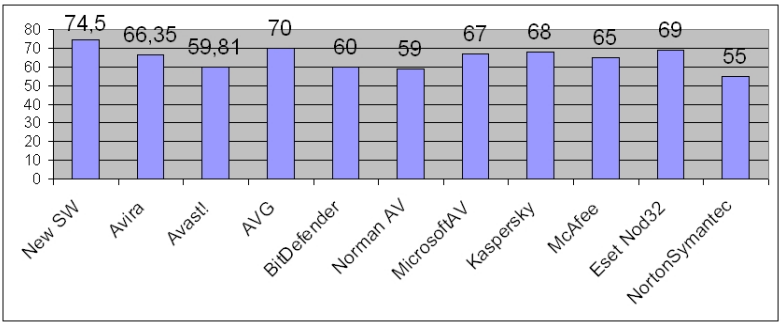


Figure 8: Reliability of Trojan detection of the developed software in comparison with known.

izes the features of Trojans functioning in computer systems is proposed. A model of the process of computer systems Trojans detection is developed. It allows performing detection with high reliability.

Software for computer systems Trojans detection was developed. It is based on the methods of detection in monitor and scanner modes and allows improving reliability and efficiency. Trojan detection software makes it possible to detect the new Trojans with high reliability and efficiency.

Notes:

[1] Mark Dowd and John McDonald, *Security Assessment: Identifying and Preventing Software Vulnerabilities* (Addison-Wesley Professional, 2006); Michael Erbschloe, *Trojans, Worms and Spyware: A Computer Security Professional's Guide to Malicious Code* (Burlington: Elsevier Butterworth–Heinemann, 2005); Szor Peter, *The Art of Computer Virus Research and Defense* (Addison Wesley Professional, 2005).

[2] Oleg Savenko and Sergiy Lysenko, "Behavioral Trojans Model," paper presented at the Computer Sciences and Information Technologies Conference, Lviv, 27-29 September 2007, pp. 129-132.

[3] Oleg Savenko and Sergyi Lysenko, "Model Search Process Trojans in Personal Computers," *Radioelectronic and Computer Systems* 7 (2008): 87-92.

[4] Oleg Savenko and Sergyi Lysenko, "Information Technology of Intelligent Trojan Detection in Computer Systems," *Radioelectronic and Computer Systems* 5 (2010): 120-126.

[5] Ruslan Grafov, Oleg Savenko, and Sergyi Lysenko, "Using Fuzzy Logic to Search for Trojan Software in Computing Systems," *Visnyk of Chernivtsi National University* 6 (2009): 85-91; Oleg Savenko and Sergyi Lysenko,"Trojan Search System using Fuzzy Inference," in *Proceedings of the Intelligent Information Analysis Conference* (Kyiv, May 2008), 413-431.

[6] Leandro Nunes de Castro and Jonathan Timmis, *Artificial Immune Systems: A New Computational Approach* (London: Springer-Verlag, 2002), 120; Stephanie Forrest , Alan S. Perelson , Lawrence Allen, and Rajesh Cherukuri, "Self-Nonself Discrimination in a Computer," *IEEE Symposium on Research in Security and Privacy*, Oakland, CA, 16-18 May 1994, 202-212; Oleg Savenko and Sergyi Lysenko, "Development of the Trojan detection process based on artificial immune systems," *Visnyk of Khmelnytskyi National University* (2008): 183-188.

[7] Sergyi Lysenko, "Adaptive Information Technology of Computer Systems Trojans Detection: Methods of Virus Detection in Computer Networks," *Visnyk of Khmelnytskyi National University* 3 (2010): 194-199.

[8] Savenko and Lysenko, "Trojan Search System Using Fuzzy Inference."

SERGIY LYSENKO graduated from the Khmelnitsky National University (Ukraine) in 2005 and received a PhD degree in 2010. He focuses on the development of information technology and Trojan detection, with more than 20 publications on the subject.

OLEG SAVENKO, PhD, is Dean of the Computer Systems and Programming faculty at the Khmelnitsky National University (Ukraine). He focuses on the development of information technologies of antivirus detection in computer systems. Author has more than 20 publications in refereed journals.