Identification of Executable Files on the basis of Statistical Criteria

Irina E. Krivtsova, Ilya S. Lebedev, Kseniya I. Salakhutdinova
Saint Petersburg National Research University of Information Technologies, Mechanics and Optics
(ITMO University)
Saint-Petersburg, Russia
{ikr, lebedev}@cit.ifmo.ru, kainagr@mail.ru

***Abstract*–The paper considers methods of identification of executable signatures using statistical criteria. Identification here should be understood as a process of file recognition by establishing its coincidence with a particular program. New ways to creation of executable file signatures are considered. A new approach to identification of elf-files based on the Chi-square and Kolmogorov-Smirnov criteria is offered. Restrictions and conditions of using these criteria are considered. The proposed method can be used to audit data-storage medium.**

I. Introduction

In this paper, identification should be understood as the process of file recognition by establishing its coincidence with a particular program. Identification of executable files is necessary when audit data-storage medium, which is an important part of information security [1].

Currently, there are various methods of file identification, for example, byte-by-byte comparison, checksum comparison and digital signature comparison [2]. Most of methods proceed from determination of integrity of their program code. However, all these methods can’t be applied to executable elf-files due to their openness, which leads to continual modification and creation of new file versions.

Unix operating systems are free (open-source software) for users, therefore they are the most convenient for a research. At that for a large variety of Linux systems (Debian, Mint, Fedora, Korora, etc.) one and the same program, after its installation, will have some differences in a code. Therefore, the development of methods for executable elf-file identification which will allow us to recognize a program regardless of its version, Linux OS on which it is installed, or the existence of minor user modifications in its code is actual task [3].

The object of study in this paper is elf-files; the subject of research is file identification; and the purpose of research is development the method of elf-file identification based on the use of Chi-squared test and Kolmogorov–Smirnov test.

Fig. 1 schematically shows the process of file identification which consists of several stages:

- at the first stage there is a signature creation for already known programs, and their adding to the archive of signatures (ARHIVE);
- at the second stage there is a signature creation for identifiable file (SIGNATURE OF FILE);
- at the third stage there is applying of statistical criteria for file identification (STATISTICAL CRITERION).

Methods to signature creation of elf-files and the archive creation were considered by authors earlier [4], [5], in this paper is implied the method of signature creation based on assembler code of a program [6].

Identification of elf-files, from the mathematical point of view, can be provided as a comparison of two frequency distributions, one of which belongs to program signature from the archive, another directly belongs to the identifiable file. A result of the comparison is interpreted as "the identifiable file is recognized as a program from the archive (result_1)" or "the identifiable file is not recognized as a program from the archive (result_2, result_3)" (Fig. 1).

The task of comparison of two frequency distributions can be solved as the task of testing statistical hypothesis by using well-known statistical criteria. Some ways of solving this problem were already considered by authors earlier [5], [7], in particular the usage of multifunction criterion Fisher's exact test (φ^*-Fischer criterion) which was applied to filtering the file signatures significantly different from the program signatures stored in the archive.

On the one hand, the task has two independent samples of assembler command frequencies (signatures), the volumes of which are determined when creating the signature, and we need to establish, if the signatures are the samples of the same distribution or not. At the same time distribution functions of the compared samples are unknown, therefore hypothesis test of samples homogeneity are applied to them [8]. In this paper the application of *Chi-squared test* (χ^2*-test*) is considered.

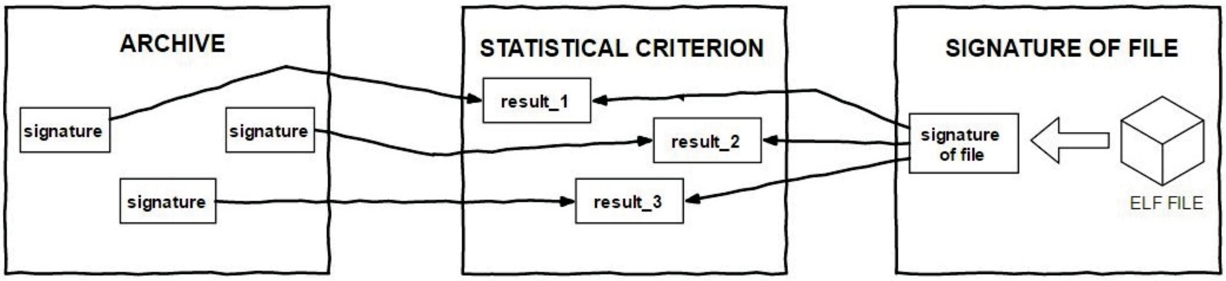

Fig. 1. File identification

On the other hand, it can be assumed that the frequency distribution of assembler commands (signature) of the program from the archive can be considered as "reference", hypothetical, while the similar frequency distribution of the identifiable file (signature) – as empirical. And we need to establish the compliance between the selective data and the hypothetical distribution that mean to apply a goodness-of-fit test. Restrictions and conditions of using *Kolmogorov-Smirnov test* are considered in this paper.

II. Identification of executable files on the basis of Chi-square test

As noted above, the identification process consists of three stages, what is shown schematically in Fig.1. Both the creation of identifiable elf-file signature and creation of program signature included in the archive are based on their assembler code [5].

A. Creation of the archive of signature

A training sample (*TS*) is formed to build the archive of signatures. *TS* consist of the elf-files, which are identified with a certain existing program and the program signature will subsequently be included in the archive. Training sample is represented as follows:

$$TS = \{v_1, v_2, \ldots, v_m\},\ i = 1 \div m,$$

where v_i – various program in an amount equal to m.

Let $v_i = \{f_1, f_2, \ldots, f_n\}$, where f_j – different versions of the i program. Each file f_j is disassembled, подсчитываются the frequencies of the most common 118 assembler commands of a program are counted $L(f_j) = (a_k)$, where $j = 1 \div n$, $k = 1 \div 118$. On the basis of these frequencies the average frequencies $\bar{a}_k$ of k-th assembler command are counted for all files f_j. As a result a tuple with 118 values is formed:

$$L(v_i) = (\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{118}),\ i = 1 \div m.$$

Next the values q_k are found by formula:

$$q_k = \begin{cases} a_k, \left|\bar{a}_k - a_k\right| \le 0,2 * \bar{a}_k \\ 0, \left|\bar{a}_k - a_k\right| > 0,2 * \bar{a}_k \end{cases},\ k = 1 \div 118,$$

and sets $Q(f_j) = (q_1, q_2, \ldots, q_{118})$ are formed. In this sets the elements of a_k which significantly different from the average $\bar{a}_k$ for the program v_i are voided. The resulting sets $Q(f_j)$ are used for the formation of the program intermediate signature $P(v_i) = (p_1, p_2, \ldots, p_{118})$ where the p_k value is calculated by the formula:

$$p_k = \begin{cases} \bar{a}_k, \sum_{j=1}^{n} \dfrac{Q(f_j)[q_k]}{Q(f_j)[q_k]} \ge 0,85 * n \\ 0, \sum_{j=1}^{n} \dfrac{Q(f_j)[q_k]}{Q(f_j)[q_k]} < 0,85 * n \end{cases},$$

in that case, if the value $Q(f_j)[q_k] = 0$, it is accepted

$$\frac{Q(f_j)[q_k]}{Q(f_j)[q_k]} = 0,$$

Eventually the program signature has an appearance:

$$S(v_i) = \{N(v_i), P(v_i)\},$$

where $N(v_i)$ – v_i program name.

All thus formed signatures are placed in the archive of signatures for the further address to it either in the course of identification, or in need of modification in signatures [5].

As for the identifiable file signature, it has the appearance:

$$S = \{P\},$$

where $P = (a_1, a_2, \ldots, a_{118})$ – frequencies the same 118 assembler commands, but in the file.

B. Use of Chi-square test

At this stage there is a direct an executable file identification by comparing the empirical distribution of 118 assembler command frequency $P(v_i)$ from the program signature stored in the archive, and the empirical distribution P the same assembler commands from the identifiable file signature. Thus, the statistical hypothesis of homogeneity of distributions is tested.

As a criterion for hypothesis tests is applied a criterion of homogeneity Chi-square, which is used both for discrete and for continuous distributions, and allows to compare the distribution of features represented in any scale starting from nominative scale [8], [9].

This criterion is quite easy to use, however, to compare distributions it is necessary to consider a restriction on the equal number of groups.

If the result of the experiment is obtained two independent samples of volumes n_1 and n_2, moreover on the considered feature the samples split into k classes with frequencies $m_1 + m_2 + \ldots + m_k$ and $m'_1 + m'_2 + \ldots + m'_k$, the empirical value χ2-test is calculated by the formula:

$$\chi^2 = n_1 n_2 \sum_{i=1}^{k} \frac{1}{m_i + m'_i} \left(\frac{m_i}{n_1} - \frac{m'_i}{n_2} \right)^2 , \qquad (1)$$

where $m_1 + m_2 + \ldots + m_k = n_1$ and $m'_1 + m'_2 + \ldots + m'_k = n_2$.

It is proved that this statistic for large values of n_1 and n_2 are distributed according to the law χ^2 with $k - 1$ the degrees of freedom [8].

It is known that the criterion of homogeneity Chi-square has right-hand critical area, therefore if in case of significance level α inequality $\chi^2 < \chi^2_\alpha$ is performed there is no reason to reject the hypothesis of homogeneity of distributions.

It should also be noted that by χ^2-criterion it is possible to check the hypothesis of homogeneity not only for two samples but also for several samples [9].

III. Identification of executable files on the basis of Kolmogorov-Smirnov test

In contrast to the method of program signature creation based on frequency of 118 different assembler commands from program disassembled code, a new approach to the study of the elf-file features is based on a choice of one assembler command accepted for formation of frequency distribution.

A. Creation of the archive of signature

Similar to the previous approach, to create an archive of signatures it is necessary to analyze a certain amount of executable files, therefore, the training sample is generated $TS = \{v_1, v_2,\ldots, v_m\}$, $i = 1 \div m$, where v_i – various program samples; m – number of different programs; $v_i = \{f_1, f_2,\ldots, f_n\}$, f_j – different versions of the i program, n – number of files in a sample.

In the beginning one of files is fixed f_{j0}, further there is its disassembling and partitioning it assembler code into intervals of unequal lengths, but with a fixed frequency of occurrence in intervals the given feature (assembler commands). In this case the quantity of different assembler commands on an interval with the fixed frequency of feature is accepted as an interval length.

The frequency distribution of feature and formed intervals for file f_{j0} are combined in the following structure:

$$L(f_{j0}) = ((a_0, h_1), (a_0, h_2),\ldots, (a_0, h_r)),$$

where a_0 – given frequency of the feature, and equal to a constant; h_i – length of the interval partitioning; r – number of intervals partitioning.

Formation of $L(f_{j0})$ is necessary in order to make possible signature creation of a program v_i with fixed number of intervals partitioning r. Further, to remaining files f_j, the following formula is applied:

$$L^*(f_j) = ((a_1, h^*_1), (a_2, h^*_2),\ldots, (a_r, h^*_r)), j = 1 \div n - 1, i = 1 \div r$$

where a_i – resulting frequency of feature; $h^*_i = \frac{h_i l^*_j}{l}$ – length of the interval partitioning, l – length of f_{j0} file, h_i – length of i-th interval partitioning of file f_{j0}, l^*_j – length of j-th file f_j; r – number of intervals partitioning.

Thus, the distributions $L(f_{j0})$ and $L^*(f_j)$ of the same length r for an program v_i are obtained, of which then a program signature is formed

$$L(v_i) = ((\bar{a}_1, h_1), (\bar{a}_2, h_2),\ldots, (\bar{a}_r, h_r)),$$

on the basis of the average values of the feature frequency:

$$\bar{a}_i = \frac{1}{n}\left(L(f_{j0})[a_0] + \sum_{j=1}^{n} L^*(f_j)[a_i] \right), i = 1 \div r, j = 1 \div n - 1,$$

As a result, the program signature takes the form:

$$S(v_i) = \{N(v_i), L(v_i)\},$$

where $N(v_i)$ – v_i program name.

All the signatures formed in this way are placed in the archive of signatures.

Peculiarity of the identifiable file signature is that their length must match the length of the corresponding program signature from the archive therefore the identifiable file signature is structured in the following way:

$$S = \{L\},$$

where $L = ((a_1, h^*_1), (a_2, h^*_2),\ldots, (a_r, h^*_r))$ – frequency distribution of assembler commands at intervals specified in the program signature from the archive; a_i – frequency of feature; $h^*_i = \frac{h_i l^*}{l}$ – length of the interval partitioning, l – sum of the lengths of the intervals h_i, specified in the signature of the corresponding program from the archive $L(v_i)$, h_i – length of the i-th interval partitioning of program signature from the archive $L(v_i)$, l^* – length of the identifiable file.

It is important to note that the proposed special method of formation of the program signature leads to the facts that:

- firstly each signature in the archive is a statistical estimate (analogue) of uniform frequencies distribution of some assembler commands. The statistical estimate is realized on the basis of the sample, which different from the sample of identifiable file frequency;
- secondly the length of each identifiable file is converted, and length of signature formed for this file is equal to the length of the program signature from the archive with which file signature is compared;
- thirdly for each signature from the archive it is needed to create the individual identifiable file signature.

B. Use of Kolmogorov-Smirnov test

The identification process is comparison of the program signature from the archive $S(v_i)$, where the frequency distribution of assembler commands is the "reference" uniform distribution, with the identifiable file signature S using Kolmogorov-Smirnov test.

In contrast χ^2-test, which compared the frequencies of the two distributions on each interval of the partitioning the Kolmogorov-Smirnov test at first compares frequencies of the first interval, further the sums of the first and second intervals, then the sums of the first, second and third intervals, etc. Thus, each time the frequencies accumulated up to the current interval are compared. The criterion allows us to find the point at which the sum of the accumulated differences between the two distributions is the greatest, and to assess the accuracy of this distance [9].

However, the Kolmogorov-Smirnov test has several restrictions, most important of which, in the context of the solved task, is the requirement to the classes' formation of the feature values. The classes must be ordered according to the increase or decrease of the feature, otherwise the accumulation of frequencies will reflect only an element of the accidental neighborhood of classes [10].

For the accounting of these requirements, as an experiment, the approach to formation of intervals partitioning of code in case of which the frequency distribution of command occurrence in an interval would have a view of uniform distribution, was considered.

The value that represents the module of the maximum difference of an empirical distribution function $F^*(x)$ from theoretical distribution function $F(x)$ is the Kolmogorov-Smirnov statistic. For a fixed volume n of sample the statistic is recorded as follows [9]:

$$d_n^* = max\ |F^*(x) - F(x)|, -\infty < x < \infty.$$

Considering that the criterion has right-hand critical area, therefore if in case of significance level α inequality $d_n^* < d_{\alpha,n}$, where $d_{\alpha,n}$ – critical value of criterion, is performed there is no reason to reject the hypothesis of distribution equality.

IV. Experiment

The experiment involved 182 elf-files of different versions and bit (32 and 64). The training sample consisted of 62 files relating to 14 programs. The test sample consisted of 123 files relating to 62 programs. Testing the efficiency of the developed methods to identify executable files based on the application of statistical criteria was the purpose of the experiment.

The experiment was conducted in two stages.

The first stage of file identification used the Chi-square criterion.

For example, let *bacula-console_5.2.5-0ubuntu6_i386* will be the identifiable file. For programs *bacula-console_i386* and *baobab_i386* were formed signatures and produced their comparison with the signature of *bacula-console_5.2.5-0ubuntu6_i386* elf-file.

Fig. 2 and Fig. 3 show for comparison the frequency distribution in the signature of identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and in signatures of program *bacula-console_i386* and *baobab_i386* from the archive accordingly. On an abscissa axis – classes of partitioning, on an ordinate axis – frequency of non-zero assembler commands of the program signature.

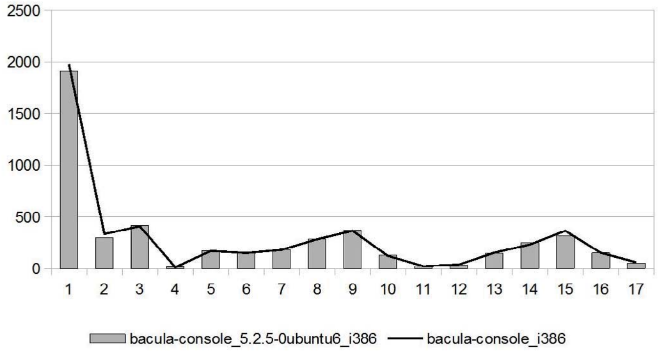

Fig. 2. The frequency distribution of the assembler commands of identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and program *bacula-console_i386* for the archive

When testing the hypothesis about the similarity of the frequency distribution in the signature of an identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and in the signature of program *bacula-console_i386* from the archive empirical value of the criterion of homogeneity $\chi^2 = 6{,}76$, which is below the critical value $\chi^2_\alpha = 26{,}30$, therefore, at the significance level $\alpha = 0{,}05$, it can be argued that the identifiable file is a program from the archive (and this is true).

Indeed, Fig. 2 shows that the histogram of frequencies of the identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and frequencies of the program *bacula-console_i386* vary slightly.

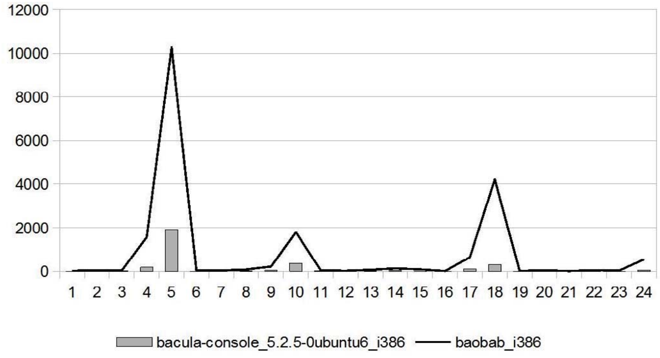

Fig. 3. The frequency distribution of the assembler commands of identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and program *baobab_i386* for the archive

When testing the hypothesis about the similarity of the frequency distribution in the signature of an identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and in the signature of program *baobab_i386* from the archive empirical value of the criterion of homogeneity $\chi^2 = 301{,}903$, which is exceed the critical value $\chi^2_\alpha = 35{,}17$, therefore, at the significance level $\alpha = 0{,}05$, it can be argued that the identifiable file is not a program from the archive (and this is true).

This conclusion is confirmed in Fig. 3, which shows graphically that the differences in the frequency distributions of identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and program *baobab_i386* from the archive are significant.

The application of the criterion of homogeneity to some other elf-files led to the results presented in Table I.

Here, the cells highlighted in dark shading corresponding to the signatures of elf-files for which empirical value χ^2-test, less than the critical value at a significance level of $\alpha = 0{,}05$. However, as can be seen from Table I, for the identifiable file *baobab_3.4.1-0ubuntu1_i386* and the corresponding program *baobab(i386)* from the archive, an error of the first type occurs (cells highlighted in light shading), when the hypothesis of homogeneity of distributions is rejected, while these distributions belong to the same program.

During the experiment, there was a need to analyze the dependence of the number of errors of first type from the sample size n in formula (1). Since the amount of information a computer program can vary from tens of kilobytes to several gigabytes, then the sample size n, i.e. the sum of frequencies of occurrence 118 assembler commands in the signature, can be from hundreds to several million. Fig. 4 shows the histogram, which displays the number of errors of the first type for different volumes of samples in the program signatures. The abscissa shows volumes of samples of 25 program signatures, the ordinate axis – the number of errors of the first type (not exceed two, because in the test sample was represented with two identifiable file for each program).

TABLE I. THE VALUES OF X^2- CRITERION

		Signatures of identifiable files			
		bacula-console_5.2.5-0ubuntu6_i386	baobab_3.4.1-0ubuntu1_i386	aptitude_0.4.11.11-1ubuntu10lucid1_amd64	b43-fwcutter_015-9 –
Signatures of programs	apt (amd64)	1002,578	2142,072	2491,678	485,242
	aptitude (amd64)	7540,314	14749,36	56,634	12769,4
	b43-fwcutter (i386)	163,916	314,279	2169,413	11,585
	bacula-console (i386)	6,755	399,749	2247,182	1851,037
	baobab (i386)	301,903	476,225	10642,139	2900,835
	bonnie++ (i386)	763,348	1228,178	8388,626	5313,883

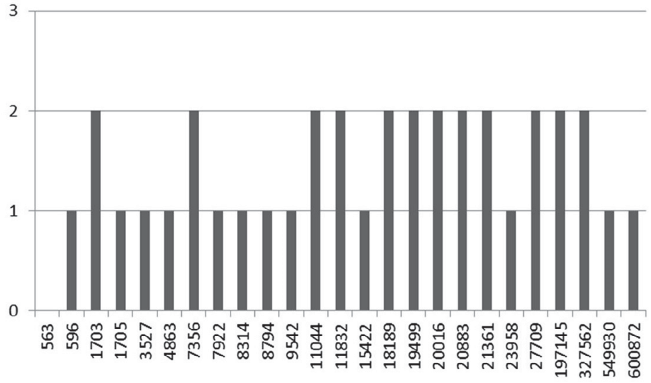

Fig. 4. The number of errors of the first type depending on *n*

It should be noted that with a large sample size n ($n > 11000$) the error of the first type occurs more frequently.

The general results of the experiment with application of the criterion of homogeneity Chi-square are presented in Table II, where the first and fourth rows contain indicators (in percentage) of correct results, in the third and fourth rows – indicators of the errors of the first and second type, respectively.

The first and second columns describe all possible outcomes of signature identification, and the third column shows the obtained results of identification for the described outcomes (in percentage).

TABLE II. IDENTIFICATION RESULTS FOR X^2-TEST

The relation between the program signature from the archive and the identifiable file signature	The hypothesis about the similarity of signatures	The result of the experiment, in percentage	
		the significance level χ^2=0,05	the significance level χ^2=0,01
The signatures belong to one and the same program	Accepted	0,36%	0,42%
The signatures belong to one and the same program	Rejected	1,23%	1,17%
The signatures belong to different programs	Accepted	0,10%	0,16%
The signatures belong to different programs	Rejected	98,32%	98,25%

The Table II shows that the indicator of correct results of identification process of elf-file signatures based on frequency distribution of 118 assembler commands is 98,68% for level of significance $\alpha = 0{,}05$ and 98,67% for $\alpha = 0{,}01$.

The second stage of the experiment for file identification was used the Kolmogorov-Smirnov criterion. As a feature for formation the frequency distributions the assembler command *cmp* was chosen.

Let *bacula-console_5.2.5-0ubuntu6_i386* will be the identifiable file again. For programs *bacula-console_i386* and *baobab_i386* were formed signatures and produced their comparison with the signature of chosen elf-file. Since the criterion requires that the sample size was large enough [6], the experiment is set value $n = 166$.

Fig. 5 and Fig. 6 show histograms of accumulation of relative frequencies for the signature of the identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and program signatures of the *bacula-console_i386* and *baobab_i386* from the archive. On an abscissa axis – classes of partitioning, on an ordinate axis – accumulation by classes.

As can be seen from Fig. 5, the accumulation of relative frequencies in the signature of identifiable file *bacula-console_5.2.5-0ubuntu6_i386* slightly deviates from the accumulation of relative frequencies in the signature of program *bacula-console_i386*. The maximum difference is $d_n^* = 0{,}073$, which is less than a critical value $d_{\alpha;n} = 0{,}106$. Therefore, on the significance level $\alpha = 0{,}05$, it can be argued that the identifiable file is a program from the archive (and this is true).

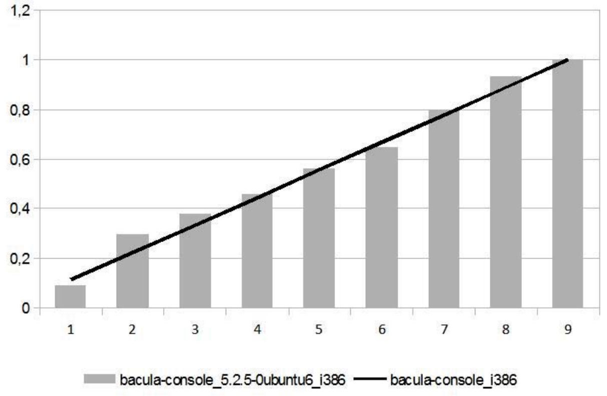

Fig. 5. Accumulation of the relative frequencies of the identifiable file *bacula-console_5. 2.5-0ubuntu6_i386* and the program *bacula-console_i386* from the archive

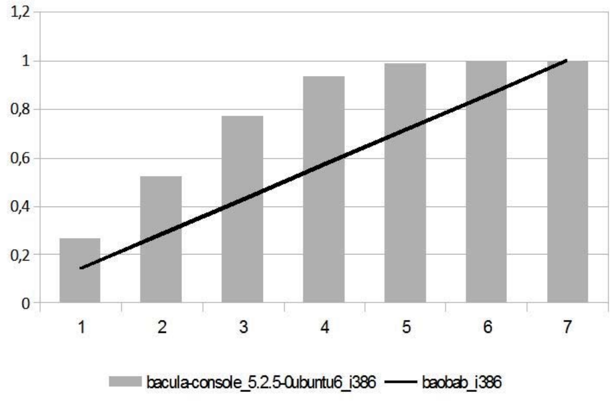

Fig. 6. Accumulation of the relative frequencies of the identifiable file bacula-console_5. 2.5-0ubuntu6_i386 and the program *baobab_i386* from the archive

From Fig. 6 follows that the discrepancy between the accumulation of relative frequencies in the signature of an identifiable file *bacula-console_5.2.5-0ubuntu6_i386* and in the signature of program *baobab_i386* is significantly; the maximum discrepancy is $d_n^* = 0{,}362$, which is more than a critical value $d_{\alpha;n} = 0{,}106$. Therefore, on the significance level $\alpha = 0{,}05$, it can be argued that the identifiable file is not a program from the archive (and this is true).

Table III shows the overall results of the experiment conducted with the use of Kolmogorov-Smirnov criterion.

The Table III shows that the indicator of correct results of identification process of elf-file signatures based on frequency distribution of one assembler command is 95,09% for level of significance $\alpha = 0{,}05$ and 93,08% for $\alpha = 0{,}01$. It can also be noticed that with the decrease in the level of significance the number of errors of the second type increases, i.e. the number of cases when the elf-file is identified as a program from the archive, while this result is wrong.

TABLE III. RESULTS OF SIGNATURE IDENTIFICATION ACCORDING TO THE KOLMOGOROV-SMIRNOV CRITERION

The relation between the program signature from the archive and the identifiable file signature	**The hypothesis about the similarity of signaturesThe level of significance $\alpha = 0{,}05$**	**The result of the experiment, in percentage**	
		The level of significance $\alpha = 0{,}05$	**The level of significance $\alpha = 0{,}01$**
The signatures belong to one and the same program	Accepted	1,86%	1,94%
The signatures belong to one and the same program	Rejected	0,61%	0,53%
The signatures belong to different programs	Accepted	4,30%	6,39%
The signatures belong to different programs	Rejected	93,23%	91,14%

V. CONCLUSION

As follows from the results of experiment presented in Tables II and III, the method of identification based on Chi-square criteria, shows a higher rate of correct results of identification of elf-file signatures. At the same time with the same level of significance ($\alpha = 0{,}05$ или $\alpha = 0{,}01$) the number of cases, when the file does not identified as a program from the archive, although the signature for the file in the archive was present (the number of errors of the first type), more than using Kolmogorov-Smirnov criteria.

However, it should be noted that for the specified level of significance, the number of cases when the file correctly identified as a program from the archive is more, if use the Kolmogorov-Smirnov test. The inevitability of creation of a large number of the identifiable file signatures for each the program signature from the archive, is can be called as disadvantage of method based on the Kolmogorov-Smirnov criterion.

In the case when the researcher needs to verify the similarity of identifiable files with a large number of program signatures it is advisable to apply the method based on Chi-square criterion. Otherwise, it is possible to apply both methods.

Thus, during the experiments the efficiency of the methods for identification of executable elf-files, regardless of their versions and the Linux operating system, on which they were installed, was confirmed.

REFERENCES

[1] F.N. Shago, I.A. Zikratov, "Technique of optimal audit planning for information security management system". *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, vol. 90, no. 2, pp. 111-117.

[2] N. Ferguson, B. Schneier, *Practical Cryptography: Designing and Implementing Secure Cryptographic Systems*. Moscow: Wiley, 2005

[3] Tool Interface Standard (TIS) Executable and Linking Format (ELF) Specification, Web: http://refspecs.linuxbase.org/elf/elf.pdf

[4] I.E. Krivtsova, K.I. Salakhutdinova, P.A. Kuzmich, "Method of Construction the Signatures of Executable Files for Identification Purposes". *Vestnik policii*, vol. 5, Is. 3, 2015, pp. 97-105.

[5] N.K. Druzhinin, K.I Salakhutdinova. "Identification of executable file by dint of individual feature" *ISPIT-2015. International Conference on Information Security and Protection of Information Technology*. St. Petersburg, Russia, November 5-6, 2015, IET - 2015, pp. 45-47.

[6] O.V. Kazarin, Theory and practice of protection programs. Moscow: MGUL, 2004.

[7] I.E. Krivtsova, K.I. Salakhutdinova, I.V.Yurin, "Method of executable filts identification by their signatures". *Vestnik gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova,* vol.1, 2016, pp 215-224.

[8] N.V.Smirnov, I.V. Dunin-Barkovskii, *Short Course of Mathematical Statistics for Technological Applications*. Moskow: Nauka, 1969.

[9] E.I. Kulikov, *Applied Statistical Analysis*. M: Goryachaya liniya-Telekom, 2008.

[10] E.V. Sidorenko, *Mathematical Methods in Psychology*. Saint-Petersburg: Rech' Publ., 2010.