Analysis of software integrity information and communication system

A A Arzumanov¹, S I Matreninskiy¹, V A Chertov¹ and D E Orlova²

¹Department of technology, construction and estate management, expertise Federal State budget educational institution of higher professional education of the Voronezh State Technical University, 84 20 October, Voronezh, RU

²ET Department, Voronezh Institute Federal Penitentiary Service of Russia, 1a Irkutskaya Street, Voronezh, RU

Email: arben1@yandex.ru, gso09@yandex.ru, cva.57@yandex.ru, dasha scorobogat@mail.ru

Abstract. On the basis of system theory and matrix management approach to the description of the structure of complex logistical systems task analysis software integrity information and communication. Describes an algorithm to detect infringements of integrity in systems of this class, both in their design and development, and in the course of the operation.

1. Introduction
The subject of research is the modeling and algorithmic software integrity analysis tools information and communication systems.

The integrity of the software is the ability of information and communication systems to ensure the immutability of circulating information in the context of accidental and/or intentional distortion (destruction) [1]. The need for such studies is due to at least two factors. First, the specificity of this class of systems is that their software components are developed by different teams of artists using a variety of technologies and programming with different views on use of software environments. Finally (and this is confirmed by the practice), with the aggregation of software products in the system, despite the agreement, there are many differences between intra-system, destroying the integrity of the final product. The second factor has to do with the fact that virtually all of the information and communication systems is forcing objects by hackers, frickers, crackers, insiders and other subjects of computer crime. Leaving (despite protection measures) in their programming environments embedded malware, destroying the structure of the software. These reasons are compelling, since software design information and communication systems, to conduct the analysis of integrity and take measures aimed at retaining it. The practical importance of this task is obvious. Experience has shown that the major costs when creating software information and communication systems linked to the analysis of its integrity and removal of internal structural contradictions created taken illegally embedded software products [2]. At present this task is considered as the art of system...
programming than strict science based on mathematical models of analysis and optimization of design solutions. As a result, it is solved by trial and error with partial-attraction assessment models for calculating individual indicators of the type of security, reliability, timeliness, accuracy, etc.

2. The formulation of the problem
Consider the conditional project to create software for some information and communication system. Let's assume that the software is a set of software components: a) developed by directly executing the project; b) purchased from third-party manufacturers; in) introduced by cybercriminals in. We assume that each of the programme components specified by a set of real or projected characteristics, namely: functions, offices, information and resource requirements. Let in the terms of reference for the development of information and communication systems integrity criteria to be met by the software design. As a rule, these criteria include [2.7]: functional and structural completeness; lack of structural and administrative duplication; information not sufficient and redundancy; resource availability and consistency; structural cohesion. The challenge will be to formalize the description of software design information and communication systems and in the development of an algorithm, allowing to check this description to match the specified criteria.

3. A formal description of software design information and communication systems
In order to develop such a description will start from the matrix approach to presenting the structure of the complex of organizational and technical systems [4], and enter into the consideration the totality of the following binary matrices that reflect the various relationships between software components that make up the software information and communication systems. [5].

\[ \bar{\alpha}_{fi}, f = 1 + F, i = 1 + I, \text{ where } \bar{\alpha}_{fi} = 0, \text{ if the execution of the f-th functions assigned to the i-th software component, } 0 \text{ – otherwise, } (1 + I) \text{ – list of software components within the software information and communication systems, } (1 + F) \text{ – general list of functions performed by the software information and communication systems.} \]

\[ \alpha_{ij}, i = 1 + I, j = 1 + J, \text{ where } \alpha_{ij} = 1, \text{ If the i-th component is managed by j-th option, } 0 \text{ – otherwise; } (1 + J) \text{ – a list of settings for the control.} \]

\[ \alpha'_{ji}, i = 1 + I, j = 1 + J, \text{ where } \alpha'_{ji} = 1, \text{ If the choice of the j-th parameter is included in the control function of the i-th component; } 0 \text{ – otherwise.} \]

\[ \beta_{ir}, i = 1 + I, j = 1 + J, r = 1 + R, \text{ where } \beta_{ir} = 1, \text{ If the i-th software component when you select the j-th parameter control uses the r-th resource, } 0 \text{ – otherwise; } (1 + R) \text{ – list of resources for the functioning of the software, information and communication systems.} \]

\[ \eta_{ii'}, i, i' = 1 + I, \text{ where } \eta_{ii'} = 1, \text{ if you select the j-th parameter provided management agreement between i-th and } i' \text{-th software components, } 0 \text{ – otherwise.} \]

\[ u'_{ijk}, k = 1 + K, \text{ where } u'_{ijk} = 1, \text{ if the i-th software component to select the j-th parameter k-th info, } 0 \text{ – otherwise; } (1 + K) \text{ – a complete list of the information needed to ensure the functioning of the software, information and communication systems.} \]

\[ u_{ijk}, K = 1 + K, \text{ where } u_{ijk} = 1, \text{ If the i-th software component when you select the j-th parameter management features K-th information, } 0 \text{ – otherwise; } (1 + K) \text{ – the full list of available information system.} \]

\[ s_{ii'}, i, i' = 1 + I, \text{ where } s_{ii'} = 1, \text{ if between i-M and } i' \text{-M components of a physical link, } 0 \text{-otherwise.} \]
4. Algorithm analysis of software integrity information and communication systems

Taking into account the imposed formalization of software integrity analysis algorithm of information and communication systems comes down to consistent checking the following conditions.

Condition of «functional completeness»: \[ \forall_{f \in F} \left( \sum_{i \in I} \bar{a}_i \right) = 1. \] If \[ \forall_{f \in F} \left( \sum_{i \in I} \bar{a}_i \right) = 0, \] the structure of software components that provide no direct performance of its functions. At \[ \forall_{f \in F} \left( \sum_{i \in I} \bar{a}_i \right) > 1 \] in the structure of the software, there are times when multiple software components perform the same function (situation in principle acceptable, but requiring special consideration because such duplication may result in a malfunction in the operation of the system).

The condition of «administrative completeness»: \[ \forall_{j \in J} \left[ \left( \sum_{i \in I} \alpha_{ij} \neq 0 \right) \rightarrow \left( \sum_{i \in I} \alpha'_{ij} = 0 \right) \right] . \] In the case where \[ \forall_{j \in J} \left[ \left( \sum_{i \in I} \alpha_{ij} \neq 0 \right) \rightarrow \left( \sum_{i \in I} \alpha'_{ij} = 0 \right) \right], \] in software present unmanaged software components, operating on their own, without control from the system. At \[ \forall_{j \in J} \left[ \left( \sum_{i \in I} \alpha_{ij} \neq 0 \right) \rightarrow \left( \sum_{i \in I} \alpha'_{ij} > 0 \right) \right] \] in the structure of the software present software components that are the objects of control by a few other software components or managed on the same issue from different places. The situation of this type are typical for software information and communication systems, which is the object of interference with the intruders, adopting viral work blocking software functional software components or taking control of them.

Condition of «information compliance»: \[ \forall_{i \in I} \forall_{j \in J} \left( \sum_{k \in K} u_{ik} u'_{jk} \neq 0 \right) . \] If this condition is not fulfilled, the structure of the software takes place either information redundancy (if \( u_{ij} = 1 \) and \( u'_{jk} = 0 \)), or information failure (if \( u_{ij} = 0 \) and \( u'_{jk} = 1 \)). In the first case the situation permissible, however, requires special consideration, because too much information can lead to a decrease in the speed of software components, implementing, for example, data stream processing. In the second case the situation intolerable, uniquely involving the failure in the operation of the software.

Condition of «handling of information»: \[ \forall_{k \in K} \left( \sum_{i \in I} u^\pi_{ik} = 1 \right) . \] Violation of this condition suggests that the software formed the unmanaged information flows, i.e. software circulates information that either nobody manages (if \( \sum_{i \in I} u^\pi_{ik} = 0 \)), one or more components manage the same information (if \( \sum_{i \in I} u^\pi_{ik} > 1 \)). Both situations are allowed but require special consideration, since both in the and in the other case, there may be glitches in software, especially in the case of information an invasion by intruders.

Condition of «resource security»: \[ \forall_{i \in I} \forall_{j \in J} \left( \sum_{r \in R} \beta_{ij} \geq 1 \right) . \] Without this, the structure of the software, there may be situations when administration produced were not supported by adequate resources. It is
obvious that the implementation of such offices is impossible. The situation of this type are typical for embedding event viral programs blocking access functional software components.

Condition of «resource approval»: \( \forall_{i \in I} \forall_{j \in J} \forall_{k \in K} (\sum_{i \in I} b_{ij} > 1 \Rightarrow (\eta_{ik} = 1)) \). If this condition is not in the structure of the software may be situations associated with the wrong choice of resource-dependent offices. Manifestation of such contradictions are man-made situations of scarce resources where such deficit is actually no, but has the wrong place (non-consensual) resource usage.

The condition of «coherence»: \( \forall_{i \in I} \forall_{j \in J} \forall_{k \in K} \left( \left( \sum_{i \in I} c_{ij} = 0 \wedge \sum_{i \in I} c_{ij}' = 0 \right) \Rightarrow \left( \sum_{i \in I} u_{i} = 0 \Rightarrow \left( c_{ik} = 1 \right) \right) \right) \). If this condition is not fulfilled, the structure of the analyzed software such that the communication needs of its software components that are not supported by the appropriate communication channels. The situation of this type are most common for software spatially-distributed information and communication systems in which communication channels exposed to deliberate interference by intruders.

In his graphic integrity analysis algorithm in the software structure of information and communication systems is presented in figure 1.

**Figure 1.** Integrity analysis algorithm in the software structure of information and communication systems.
5. Conclusion
The algorithm analysis of the integrity of the software structure of information and communication systems for various purposes. As a result of its use can be identified violations of the integrity of the eleven types, namely: "functional incompleteness", "functional duplication", "management incompleteness" of "structural duplication", "management duplication", "information redundancy", "information insufficiency", "unmanaged information flows", "lack of resource", "inconsistency in resource", "connectivity violations". The dignity of the algorithm is that it allows real-time solve tasks of identifying bottlenecks in software information and communication systems for various purposes, as well as in the stages of their design to select structures meet the requirements of the functional and structural completeness of information without redundancy and adequacy, resource availability and consistency, the lack of structural and administrative duplication and connectedness. When implemented in a programming language, the algorithm can be used as a software component that provides integrity monitoring software information and communication systems in conditions of contradictions with hackers, insiders, frickers, crackers and other subjects of computer crime [6].

References
[1] Russia Gostekhkomissija guidance document: protection from unauthorized access to information 1992 Terms and definitions (Moscow: military Publishing House) p 134
[2] Demin B E 2006 Methodological bases and models of large-scale projects in information and communication systems (Voronezh: scientific book) p 332
[3] Polyanichko M A 2013 Technique of automated diagnostics complex conflicts in software protection of information computer system: the thesis is a candidate of technical sciences (Sbp: St. Peterb) p 135
[4] Novoseltsev V I and Tarasov B V 1988 Logico-linguistic models in military system studies. (M.: Voenizdat) p 280
[5] Novoseltsev V I and Tarasov B V 2011 Systems theory to the conflict (M: Mayor) p 336
[6] Balan V P, Dushkin A V and Novoseltsev V I 2015 Conflict Management: a training manual for universities (Moscow: Hot line – Telecom) p 144
[7] Clark D and Wilson D A 1987 Comparison of Commercial and Military Computer Security Policies Proce Symposium on Security and Privacy (Oakland Cal) p 154