International Journal of Safety and Security Engineering
Vol. 13, No. 6, December, 2023, pp. 1039-1047
Journal homepage: http://iieta.org/journals/ijsse

Development of a Method for Determining the List of Key Threats to Information Security of Computer Networks

Abas Lampezhev, Vladimir Kuklin, Leonid Chervyakov, Aslan Tatarkanov*

Institute of Design and Technology Informatics, Russian Academy of Sciences, Vadkovsky Lane 18 building 1A, Moscow 127055, Russia

Corresponding Author Email: as.tatarkanov@yandex.ru

https://doi.org/10.18280/ijsse.130606

Received: 11 October 2023
Revised: 16 November 2023
Accepted: 29 November 2023
Available online: 25 December 2023

Keywords:
computer network (CN), metagraphs, method, threat modeling

ABSTRACT

In the field of information system (IS) security, a comprehensive enumeration of potential threats remains a formidable challenge. This study introduces a novel methodology designed to amalgamate the majority of known threats, taking into account the intricate interactions among the various internal components of an IS. The foundation of these assemblages is shown to be rooted in metagraph theory, with a detailed examination of the corresponding construction models provided. The core principle of the proposed method involves the generation of an expansive threat inventory, derived from an analysis of the interrelationship matrix predicated on a computer network model. An evaluation of this method indicates that it successfully addresses previously identified deficiencies, leading to a substantial augmentation of the roster of critical threats and, consequently, a marked enhancement in the security metrics of the computer network (CN).

1. INTRODUCTION

Cybersecurity (CS) for computer networks (CNs) has remained a persistent and evolving concern from the inception of CNs to the present day [1]. A critical component of CS algorithms is the development of a comprehensive list encompassing all contemporary threats [2]. Despite ongoing advancements, the majority of safety issues pertinent to CS continue to be pressing [3].

A paramount objective in advancing CS methods is to diminish the dependence on the expertise of specialized personnel in the compilation of a definitive list of key threats. This research is directed toward the establishment of an innovative method for assembling such a list for the CS of CNs. Distinct from existing approaches, the proposed method employs specialized matrices that delineate communication protocols between pairs of elements. These matrices articulate the interactions among internal elements that replicate the CN and models that represent threats to CS.

The research has yielded several noteworthy outcomes:

1. The construction of a CN model utilizing metagraphs, which stands apart from similar constructs due to the delineation of relationships among diverse software components within the CN.
2. The formulation of a threat model to CS for any CN, characterized by the generation and categorization of various threat scenarios, underpinned by the analysis of metagraphs.
3. The introduction of a novel method for enumerating key threats to the CS of CNs, which is distinguished from analogous methods by the utilization of matrices that explicate the interactions of internal components.

The utility of this approach is further emphasized by the employment of attribute metagraphs, which are noted for their capacity to map connections across different software strata of computer systems.

2. LITERATURE REVIEW

Contemporary methodologies for modeling threats to cybersecurity (CS) span an array of practices, encompassing established regulations and standards, documented experiments, and both applied and theoretical scientific investigations [4]. In the endeavor to simulate threats and construct models that mirror the behavior of adversaries and assess protection levels, a precise depiction of the computer networks (CNs) under consideration is imperative. It is posited that an exhaustive representation of an information system (IS) enhances the likelihood of identifying the maximum array of threats [5].

The simulacra of CN behavior, albeit somewhat abstracted, can be categorized into several classifications [6], including:

- conceptual;
- functional;
- mathematical.

Goel and Chen's research [7] advocates for a nuanced representation of CN threats, necessitating a suite of models that emulate the conduct of each constituent and their interrelations. Such an approach is advanced on the premise that a singular, initial model is deficient in accurately mimicking the complex behavior of a CN. This study has thus elected to anchor its methodology in graph theory, a

mathematical model renowned for its fidelity in portraying structures analogous to CNs.

The study of a mathematical model that simulates a CN revealed several of its shortcomings [8]:

− It is quite difficult to establish the algorithms by which each component operating in the IS is guided. The nature of their relationship with each other has not been fully clarified.

− IS simulation is based on oriented graphs that describe the nature of communication of CN components at several OSI levels: real and network. Undoubtedly, not every level of the ideal OSI model was considered because the level of software and operating system (OS) interrelationship in CN was not considered here.

− Modeling reflects the degree and nature of the relationship between the "sphere of threat", "the sphere of the protection system", and the "sphere that needs to be protected". The model does not imply detailed descriptions of the communication of any of its components belonging to the protected area. Moreover, there is no detailed description of each of its components.

The endeavor to compile an exhaustive list of key threats to cybersecurity (CS) is frequently hampered by the sheer volume of information, or rather the lack thereof, pertaining to each potential threat. Researchers are often constrained by their capacity to identify and categorize the burgeoning number of new threats [9, 10]. Even with a hypothetical complete inventory of key threats at their disposal, the processing thereof could be overwhelmingly time-intensive. Consequently, reliance on a variety of methodologies for the classification and identification of emergent threats is advocated [9, 10].

Hettiarachchi and Wickramsinghe [11] conducted assessments of several threats to organizational information systems, proposing a spectrum of remediation strategies facilitated by strategic decision-making and the integration of awareness algorithms. Furthermore, Nicho and Kamoun [12] have advanced predictive simulations aimed at equipping IS professionals with the means to preemptively address internal security challenges. Complementing these efforts, Samson and Usman [13] have appraised a range of techniques to sustain an adequate level of security within computer networks (CNs). These techniques are recognized as instrumental in safeguarding the resource base of CNs and the integrity of the data transmitted therein.

A thorough analysis of various models simulating threats to CNs has revealed significant limitations:

− The heterogeneity observed in the depiction of potential threats presents substantial challenges for researchers endeavoring to model a definitive list of key threats that are applicable to actual CNs.

− Certain models are found to incorporate actors who may exploit vulnerabilities, as exemplified by potential breaches of consumer trust in the context of cloud services.

− The absence of a distinct delineation between information threats and those that pose a grave risk to the operational integrity of CNs has been noted. Without such classifications, the selection of appropriate protective measures becomes arduous, thereby complicating the establishment of a robust defense framework [14].

− The expertise of specialized personnel has been identified as frequently insufficient.

3. METHODS AND EFFECTIVENESS OF ASSESSING

3.1 Techniques to ensure the compilation of a list of key threats to the information security of each CN

Based on the results of the assessment, which allows modeling of each threat, and techniques that analyze the degree of risk to which IS are exposed, we can conclude that a key list of threats can be obtained:

– By describing the system and defining the totality of its components that need to be protected.

– By determining the degree of threat to each component identified at the 1st stage of creating the list.

Subsequently, techniques should be applied that compile a list of key threats, based on a matrix describing the relationship between its components.

3.2 Generalized mechanism for method functioning

The method that opens up the possibility of compiling a list of key threats to CS covers several stages:

– Classifying each component belonging to the CN.

– Creating a matrix that reflects the nature of the relationship among all components.

– Providing a list of key threats.

The order of operation of the method presented in IDEF0 is shown in Figure 1.

The amount of input data should include the following:

– Lists of additional and systemic software packages.

– Specifics of the logical structure of the assessed CNs.

– Lists of protocols used to ensure the stable transmission of information.

– Several requirements regarding the consideration of threats to the level of security in the general list of threats.

The practical implementation of this method makes it possible to form a complete list of threats to the CSs of the considered CNs.

The control mechanisms include the imitation of the CN behavior, which is described below.

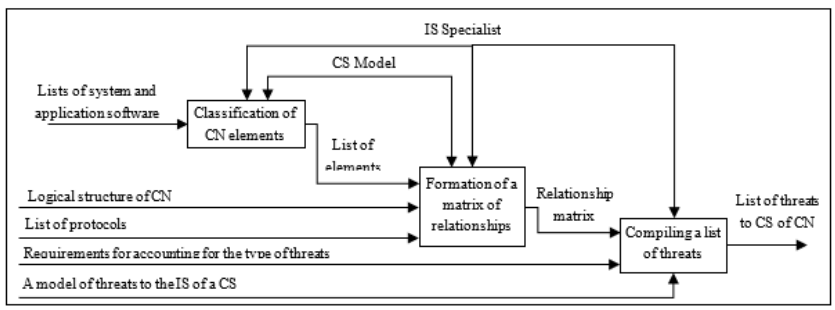

Figure 1. Representation of the methodology for compiling a list of key threats to the CS of a CN

It is necessary to clarify some initial data. Thus, CNs are studied in the format of a structure that includes interconnected components, and the available internal relationships among them are also assessed.

Threats refer to the category of unauthorized amendments recorded in any CN. Therefore, every threat affecting security, rather than the information environment, is subject to consideration.

Today, when discussing accessibility, the degree of information accessibility should be considered, rather than the level of accessibility of the average user.

It should also be noted that at this stage, the degree of interrelationship of each component is assessed. It turns out that the classification of the components that form any CN can be modeled by graph vertices. This figure shows the following:

1. available software;
2. reliable OS;
3. a number of subnetworks.

It is assumed that each protocol and component included in the CN fully corresponds to each layer of the OSI models:

– The software must function on the basis of the protocols that are at the application level and on the relationships of the OS level.

– The OS will interact on the basis of protocols of various levels.

– Any subnetwork will interact on the basis of protocols of various levels.

– The relationship matrix, which is the basis of our method, can be expressed as follows:

1. Identify a list of components included in the CN regarding their compliance with the logical structure of the CN.
2. Identify a list covering all involved protocols.
3. Compare lists of components and protocols.

When delving into the essence of what is happening, the list of components enters the input channel of the method being developed, and it is necessary to determine the degree of correspondence and belonging of each component to the general hierarchy: Subnetworks include operating systems, and they cover all software.

A list combining protocols is created from each protocol, due to which the communication of components within certain CNs is performed.

By comparing lists, it is possible to establish those protocols, relying on which pair of components interact. At the same time, each stage of creating a matrix is divided into a number of successive steps. Guided by the description of the method, it can be concluded that several requirements are put forward for the models used in the matrix that simulate the behavior of the CN, which consider the following:

– Software hierarchy of a particular CN.

– Probability of having a set of relationships among its components.

– The components and the relationships that unite them are described by certain characteristics.

3.3 Metagraphs

The potential of the mathematical complex of metagraphs is the basis of the model that simulates the behavior of the threat and the behavior of the CN. It is necessary to use the provisions put forward by the theoretical foundations of graphs during detecting any threat, as confirmed by Sony and Naik [15] and Godquin et al. [16].

The metagraph allows for the harmonization of two system characteristics: integrity and the possibility of division. For this reason, a subsystem can be derived from the systems, which opens up the possibility to consider the system, or its subsystem, guided by what currently interests the researcher.

Basu and Blanning [17] define a metagraph as a special summation, where ordinary graphs and hypergraphs act as summands. Wang et al. [18] considered the problems associated with metagraphs that arise during visualizations. The researchers also proposed a universal algorithm capable of implementing a software tool for simulating a particular CN. For metagraphs, Šajna and Wagner [19] provide a definition of a metavertex, which represents a certain union that includes several vertices.

Metagraph edges connect the two metavertices. Each edge can have differences in its qualities; therefore, metagraphs can be considered as a type of multigraph (Figure 2).

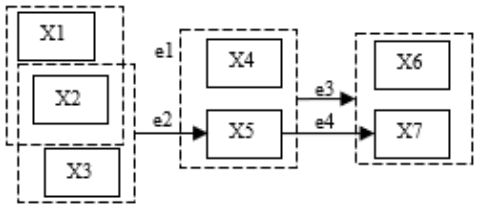

Figure 2. Metagraph representation

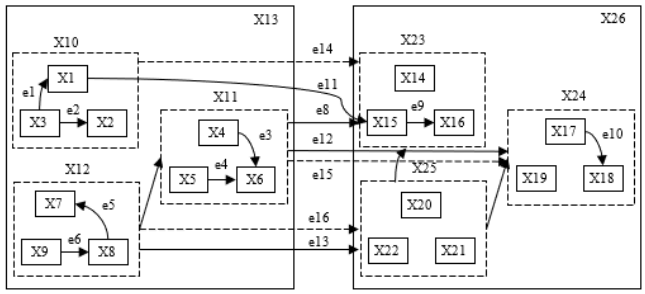

Figure 3. Metagraph of nests *n*

Godquin et al. [16] proposed the determination of the metagraphs of nests *n*, which appear as follows (Figure 3):

$$G = (X, E), \tag{1}$$

where, *G* is a metagraph of nests *n*;

$X = \{x_i\}, i = \overline{1,n}$ are non-empty incomplete sets covering the vertices;

$E = \{e_k\}, k = \overline{1,m}$ – non-empty incomplete sets covering graph edges.

Any edges of the *n*-dimensional graphs connect several subsets related to a set of vertices:

$$e_k = (V_i, W_i), \tag{2}$$

where, $V_i, W_i \subseteq X$; $V_i \cup W_i \neq \emptyset$; *i* is the level of nests.

At the same time, there are the following dependencies:

$$f_1^l: g_1^l(x_1^l, e_1^l) \to x_2^p, f_2^p: g_2^p(x_2^p, e_2^p) \to x_3^m, \dots, f_{n-1}^t: g_{n-1}^t(x_{n-1}^t, e_{n-1}^t) \to x_n, \tag{3}$$

where, *l,p,r,…,t* are the numbers of each vertex and edges of certain levels.

Basu and Blanning [17] describe the characterizing metagraph. Here, each edge or vertex can have many

characteristics. In the text of this research, we use exceptional metagraphs, but today there are the results of some assessments that deepen the provisions of theoretical foundations that describe the graphs. For example, Hinding et al. [20] offer and describe in detail the varieties of summation of graphs expressed by protographs and archigraphs.

Such metagraphs can be represented by an orderly four:

$$MG = (V, MV, E, ME), \quad (4)$$

here, *MG* is a metagraph; *V* – sets that combine the peaks of metagraph vertices; *MV* – a set that includes metagraph metavertices; *E* represents the totalities covering the metagraph edges; *ME* – the totalities that cover the metagraph metaedges.

Metagraph vertices are inherent in the totality of characteristics:

$$v_i = \{atr_k\}, \quad (5)$$

here, v_i is a metagraph vertex, $v_i \in V$;

atr_k is the current characteristic.

The metagraph edges have a totality of characteristics, vertices, and several signs of direction:

$$e_j = (V_s, V_E, eo, \{atr_k\}), \quad (6)$$

here e_j is a metagraph edge, $e_j \in E$; V_s – a metavertex of one of the edges; V_E – a metavertex of one of the edges; *eo* – a sign that displays the orientation of the edges; atr_k – a current characteristic.

Thus, the edge directionality features can be characterized by the following values:

$$eo = true|false, \quad (7)$$

where, *eo* = *true* displays directed edges; *eo* = false displays a number of non-directed edges.

3.4 CN models

Currently, most ISs rely on several functional stations that are joined in local area networks (LANs). Any functional station in the LAN has an operating system (OS) that includes several software packages [21, 22]. The functioning of the model that imitates IS behavior is based on the method proposed by Shelupanov et al. [23].

The inputs of the model are sets of software, operating system and LANs. The main tasks of this model are to describe the information system, using the structure of the attribute metagraph of nesting 3, and to form links between the given input elements (between software, operating systems and local area networks). The links will be described according to the OSI reference model.

Thus, the OSI model assumes that information among network objects corresponding to different levels will be transmitted under the guidance of several protocols operating for these objects [24].

At the same time, the real and session levels will focus on interaction with the software; therefore, such a relationship can be simulated by protocols at the OSI levels. However, this series of protocols will not be identical to the OSI model in every case, which means that the dependencies of the session levels will be added to the practical ones.

A number of protocols corresponding to the transport levels can be implemented using software tools – OS components. Therefore, any communication in the OS can be described by protocols operating at the OSI transport levels.

Every communication in a LAN environment is performed by routers that use the protocols of the OSI network layers; therefore, the interrelationship in the LAN environment is represented by the protocols that run on these OSI layers [24].

We will not consider the abovementioned levels because this research evaluates IS behavior in virtual environments, and the abovementioned levels reflect the degree of interrelationship of components in real environments.

Each layer describing the relationship of the IS components according to the OSI model is shown in Figure 4.

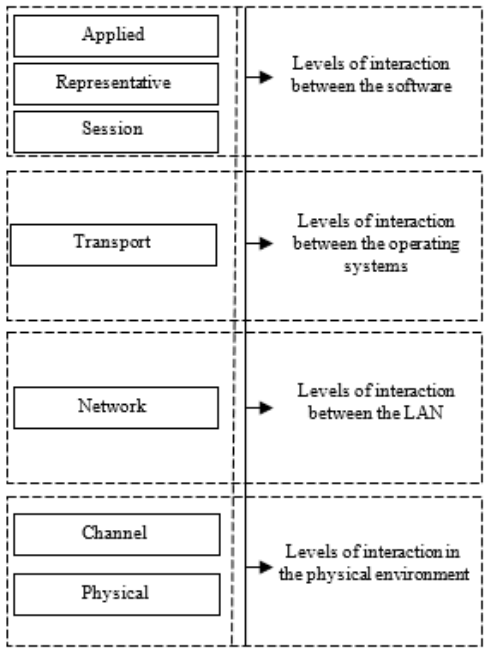

Figure 4. Levels of the interaction of the IS components

Any information system is a collection of interdependent components. Therefore, the system is only one of the components of the complex belonging to the higher hierarchy, and any of its elements belong to secondary systems.

IS can be described by nesting metagraph 3, which is built and described in detail by Basu and Blanning [17].

Since this research evaluates only the CN software component, we are dealing with a conflict between the definitions of "operation system" and "software". More appropriate phrases are "system software" and "application software". Therefore, we introduce several definitions that describe levels:

- Application software is the software layer.
- System software is the OS level.
- The software that organizes the transfer of information to the CN is a certain level of the subnet.

Nesting metagraph 3 can be the following tuple [17]:

$$G = (X_1, X_2, X_3, E_1, E_2, E_3), \quad (8)$$

where, G is nesting metagraph 3; $X_1 = \{x_1^k\}, k = \overline{1,q}$ represents sets of software packages; $X_2 = \{x_2^l\}, l = \overline{1,r}$ represents the sets covering OS, $x_2^l \subset X_1$; $X_1 = \{x_1^k\}, k = \overline{1,q}$ – LAN sets, $x_3^m \subset X_2\$\ E_1 = \{e_1^n\}, n = \overline{1,t}$ – collections that display relationships in the software environment, based

on a certain amount of X_1; $E_2 = \{e_2^o\}, o = \overline{1,u}$ - collections that display relationships in the OS environment, which are determined by the set X_2; $E_3 = \{e_3^p\}, p = \overline{1,v}$ – collections that display relationships in the LAN environment, which are determined by the set X_3.

Table 1. Characteristics of each element of the set

An Element of the Set	Characteristics
X_1 (the software set)	1. Software name 2. Numbers of the software driver versions 3. Software location 4. Numbers of ports used by the software
X_1 (the OS set)	1. OS name 2. IP addresses used by the OS 3. View of the current OS settings
X_1 (the LAN collection)	1. LAN name 2. MAC addressed of network devices 3. The name of the protocol according to which the interrelationship is performed in the LAN environment 4. Setting the protocol operating in this CN
X_1 (collections covering relationships in the software environment)	The name of the protocol related to the applied and session OSI layers
X_1 (collections covering relationships in the software environment)	The name of the protocol belonging to the OSI transport layer
X_1 (collections covering relationships in the software environment)	The name of the protocol belonging to the OSI network layer

Concurrently, the following dependencies should be considered:

$$f_1^w: g_1^w(x_1^k, e_1^n) \to x_2^l, \tag{9}$$

where, x_1^k – an element belonging to the software set.

e_1^n – an element belonging to the collection that displays the relationships in the software environment; x_2^l – an element belonging to a set covering all OSs.

$$f_2^y: g_2^y(x_2^l, e_2^o) \to x_3^m, \tag{10}$$

here, x_2^l is an element corresponding to the set including OS; e_2^o – an element that belongs to the collection representing the relationships in the OS environment; x_3^m – an element belonging to the LAN set.

Vertices are described by sets of characteristics:

$$x_i^b = \{atr_a\}, \tag{11}$$

here, $i = \overline{1,3}$ are vertex nesting levels; b – numbers of vertices belonging to certain levels i; atr_a – characteristics of the vertices (numeric, line, etc.).

Edges can be described by the following set of characteristics:

$$e_j^h = \left(x_j^c, x_j^d, \{atr_z\}\right), \tag{12}$$

here, x_d^c is the initial vertex of the edges; x_j^d – the final vertex of the edges; $j = \overline{1,3}$ – edge nesting level; atr_z – the current characteristic of the edges (numeric, line, etc.); h – edge numbers at certain levels j; c, d – vertex numbers at certain levels j.

Some possible characteristics of the components of each considered set are given in Table 1.

3.5 Threat models to CN cybersecurity

3.5.1 Threat models that face CN integrity

The created model contains a list of key threats to the integrity of IS components [25].

Threats can take several forms:

– Threats to the totality of components (metagraph vertices);

– Threats to a collection covering relationships in the environment of components (metagraph edges).

Moreover, any attack can comprise several attacks.

Threats to a combination of components:

1) Component substitutions (vertex).
2) Component exceptions (vertex).
3) Component additions (vertex).

Threats to the collection covering relationships:

1) Relationship substitutions (edge).
2) Relationship exclusions (edge).
3) Relationship additions (edge).

The simulation of the threat to the integrity of the IS components is composed of the sum of each C_{si}:

$$C_s := \Big(C_{s1}(x_1^k x_2^l x_3^m) \cup C_{s1}\left(e_1^n e_2^o e_3^p\right) \cup C_{s2}(x_1^k x_2^l x_3^m) \cup C_{s2}\left(e_1^n e_2^o e_3^p\right) \cup C_{s3}(x_1^k x_2^l x_3^m) \cup C_{s3}\left(e_1^n e_2^o e_3^p\right) \cup C_{s4}(atr_a) \cup C_{s4}(atr_z)\Big), \tag{13}$$

here, C_{s1} is a threat class that replaces a component or relationship:

– $C_{s1}(x_1^k x_2^l x_3^m)$ – a threat capable of replacing a component (vertex) at certain levels;

– $C_{s1}\left(e_1^n e_2^o e_3^p\right)$ – a threat associated with the substitution of relationships (edge) at certain levels;

– C_{s2} is a class of threat associated with the removal of a component or relationships:

– $C_{s2}(x_1^k x_2^l x_3^m)$ – a threat associated with the removal of a component (vertex) at certain levels;

– $C_{s2}\left(e_1^n e_2^o e_3^p\right)$ – a threat associated with the removal of relationships (edge) at certain levels;

– C_{s3} is a class of threat associated with the addition of a component or relationships;

– $C_{s3}(x_1^k x_2^l x_3^m)$ – a threat associated with the addition of a component (vertex) at certain levels;

– $C_{s3}\left(e_1^n e_2^o e_3^p\right)$ – a threat associated with the addition of relationships (edge) at certain levels;

– C_{s4} is a class of threat associated with changes in the settings of a component or relationships;

– $C_{s4}(atr_a)$ – a threat associated with changes in the settings (characteristics) of the component (vertex);

– $C_{s4}(atr_z)$ – a threat associated with changes in the settings of relationships (edge).

The indicators of the existing relationships can be set by the vertex indicator. Some characteristics that describe the relationships in the software collection are represented by the port numbers used in the software. At the same time, the OS collection is expressed by the information protection (IP) addresses used by the OS. Indicators describing the

relationships of LAN sets are expressed by the IP addresses of the networks and the routing tables used for interconnection in the environment of the components belonging to this CN.

The collection of threats associated with the integrity of the IS components is given in Table 2.

Each collection of software, OS, and LAN can be designated as X_1, X_2, X_3. The collection covering the relationships in the software, OS, and LAN environments is denoted as E_1, E_2, E_3. The collection of characteristics software, OS, and LAN will be denoted as atr_a. Here, $atr_a \in x_1^k$ – will correspond to the software level, $atr_a \in x_2^l$ – will correspond to the software OS level, and $atr_a \in x_3^m$ will correspond to the LAN level.

Table 2. Collection of threats related to the IS integrity

	LAN Sets	OS Sets	Software Sets
Threat to the set of vertices	Substitutions, exclusions, and additions		Substitutions, removal, and installation
Threat to the set of vertex characteristics	Changes in the routing tables or IP addresses of networks	Changes in IP addresses used by OS	Changes in the number of ports used by the software
Threat to a set covering all edges	Protocol substitutions, removal, and additions		
	Operating at the LAN levels	Operating at OS levels	Operating at the software levels

3.5.2 Modeling threats to CN privacy

This research could simulate the privacy threat behavior of IS components, given the suggestions of Konev [25]. At the same time, the IS structure was used, which was built due to the metagraph.

With a full-fledged IS, we have described several threats that apply to any vertex and to any edge of metagraphs. Several threats to the characteristics of the metagraphs were described.

The main indicators of metagraphs are as follows:

– OS version numbers.

– Versions of the protocols that provide information transmission.

– Name of the workstations.

– IP addresses.

– The lengths of the keys used to encrypt.

Threats can be expressed as:

– Disclosing data about any protocols according to which OS interconnection is organized in the corresponding LANs.

– Disclosing information about the name of the software operating in any OS.

– Disclosing information about any port used to transfer information in a workstation environment, etc.

The generated threat behavior model describes several threats to the privacy of the IS components.

Privacy displays a data characteristic that restricts third-party access to this information.

Considering confidentiality in a LAN, violators are interested in the entire amount of information about the OS and software installed in this OS, as well as a number of protocols, with regard to which the interconnection of CN components is implemented, including their settings. If attackers become the owners of information about the configuration of any LANs, they will be able to harm such networks.

Creating a model is an addition to simulating threats to the integrity of IS components. The same information leaks that reveal data about any vertex can be attributed to threats to sets of characteristics.

Threats to the set covering edges can include information leaks about any edge.

A model simulating threats to CN privacy – K_s will consider all K_{si}:

$$K_s := \left(K_{s1}(x_1^k x_2^l x_3^m) \cup K_{s1}\left(e_1^n e_2^o e_3^p\right) \cup K_{s2}(atr_a) \cup K_{s2}(atr_z)\right), \quad (14)$$

here, K_{s1} is a threat class associated with the disclosure of information about the name of the component and relationships:

– $K_{s1}(x_1^k x_2^l x_3^m)$ – a threat related to the disclosure of information about the name of the components (vertex) at the appropriate level;

– $K_{s1}\left(e_1^n e_2^o e_3^p\right)$ – a threat related to the disclosure of information about the name of the components (edge) at the appropriate level;

– K_{s2} is a threat class associated with the disclosure of information about the setting of components and relationship:

– $K_{s2}(atr_a)$ – a threat related to the disclosure of information about setting the characteristics of components (vertex);

– $K_{s2}(atr_z)$ – a threat related to the disclosure of information about the relationship settings.

Each setting can be set in the vertex characteristics. Relationships can be set in the characteristics of any edge. The settings here include all information about the software versions, OS versions, and each driver that uses the software. Network settings include information about each protocol, such as key lengths that are used for encryption, protocol versions, max-address, and IP address of each network device, etc.

At the same time, the level of damage is determined by the importance of the processed data in the IS.

The name of the software, OS, and LAN is determined by the full addresses of the CN components, including their full names.

The collection of threats to the set of vertices is given in Tables 3 and 4.

The totality of threats to a large number of characteristics of any vertex is given in Table 5. The totality of threats to the disclosure of information regarding network settings for the collections of each vertex is given in Table 6.

Table 3. Collection of threats to the privacy of multiple components

Sets	Threat to CN Privacy
X_3 (LAN sets)	LAN name disclosure
X_2 (OS sets)	OS name disclosure
X_1 (Software sets)	Software name disclosure

Table 4. Collection of threats to the privacy of collections covering relationships

Collections	Threat to CN Privacy
E_3 (collections covering relationships in the LAN environment)	Disclosure of information about the protocols according to which the interconnection of LAN components is organized
E_2 (collections covering relationships in the OS environment)	Disclosure of information about the protocols according to which the interconnection of OS components is organized
E_1 (collections covering relationships in the software)	Disclosure of information about the protocols according to which the interconnection of software components is organized

Table 5. Collection of threats to the privacy of each setting

Collections	Threats to IS Privacy
$\{atr_a\} \in x_1^m$ (set of LAN characteristics)	Disclosure of information about the LAN settings
$\{atr_a\} \in x_1^l$ (set of OS characteristics)	Disclosure of information about the OS settings
$\{atr_a\} \in x_1^k$ (set of software characteristics)	Disclosure of information about software settings

Table 6. Collection of threats to privacy for a large number of characteristics of each edge

Collections	Threat to CN Privacy
$\{atr_z\} \in e_3^p$ (collections of characteristics describing relationship in the LAN environment)	Disclosure of information about protocols at the LAN levels
$\{atr_z\} \in e_2^o$ (collections of characteristics describing relationship in the OS environment)	Disclosure of information about protocols at the OS levels
$\{atr_z\} \in e_1^n$ (collections of characteristics describing relationship in the software environment)	Disclosure of information about protocols at the software levels

3.5.3 Models that simulate threats to CN integrity and confidentiality

The security of T_S systems is determined by the union of K_S, and C_S:

$$T_S = K_S \cup C_S; \quad (15)$$

here, K_S is a model of threat to the privacy of IS components; C_S – a model of threat to the integrity of information system components. By combining the collections, a set will be created that includes all the components belonging to the 1st or 2nd sets.

Therefore, the overall CN security level is represented by the sum of the following equations:

$$\begin{gathered} T_S := \left(C_{s1}(x_1^k x_2^l x_3^m) \cup C_{s1}(e_1^n e_2^o e_3^p) \cup \right. \\ C_{s2}(x_1^k x_2^l x_3^m) \cup C_{s2}(e_1^n e_2^o e_3^p) \cup C_{s3}(x_1^k x_2^l x_3^m) \cup \\ C_{s3}(e_1^n e_2^o e_3^p) \cup C_{s4}(atr_a) \cup K_{s1}(x_1^k x_2^l x_3^m) \cup \\ \left. K_{s1}(e_1^n e_2^o e_3^p) \cup K_{s2}(atr_a) \cup K_{s2}(atr_z)\right) \end{gathered} \quad (16)$$

Thus, two models based on metagraphs were developed and combined: the model of integrity threats and the model of confidentiality threats of information system elements, into one, which is called the model of threats to the security of the computer network.

4. ASSESSMENT RESULTS

We found that owing to the proposed approach, which allows simulating threats, specialists involved in information data protection can consider eleven options for threats to the CS of any CN in the course of building security complexes. In total, based on the authors' classification, 36 options of threats to the integrity of systems were identified, while the data bank containing information on threats [26] offers only 25. This list includes threats to the software, operating system, and subnetwork levels.

During practical use of this assessment, we have compiled a list covering seventy threats to the integrity of the system. Each threat was studied at several levels with respect to all key components of the CN. The result of the practical application of this method was a list that overcame the list compiled by the client's specialists by 17%, which is presented in Figure 5. This circumstance opens up the possibility of considering such threats during designing complex protection by introducing auxiliary protection tools. In addition, the authors' model that simulates IS makes it possible to consider each characteristic of the components and their interrelationship.

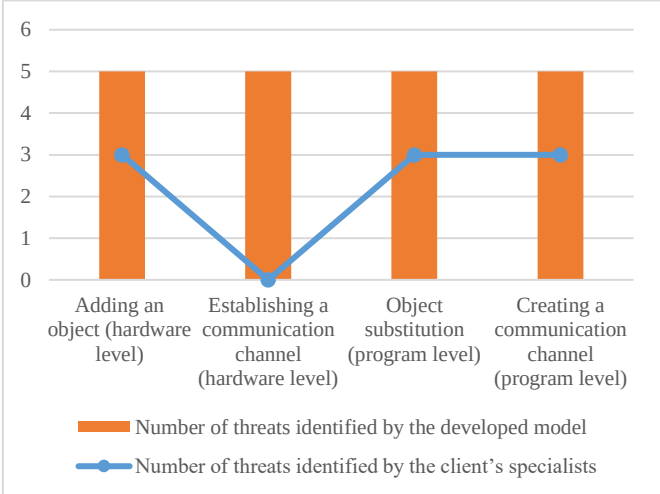

Figure 5. Comparison of threat detection methods

5. CONCLUSION

As a result of the assessment, a method was created to determine the list of key threats, and two models were presented, one that simulates the CN and another that simulates the behavior of each threat to the CN.

Its main advantage is repeatability. For example, its use by different specialists leads to a similar list of threats, regardless of the qualifications of these specialists. In addition, this method can be used as one of the steps in assessing the security level of a CN. However, the condition in this case is the correct formation of the list of elements of the system under consideration.

Simulation of each threat, based on working with metagraphs, opens up the prospect of compiling complete lists of threats to CN privacy. It should be noted that the created model, which simulates the behavior of a threat, represents most threats more qualitatively than the threat data bank,

which stores information about all known threats. This model could identify 11 more threat options.

A number of shortcomings that occurred when simulating threats were considered and eliminated. For example:

1) Components belonging to the violator model were practically excluded.

2) Each threat was described in detail and classified.

3) Each threat was assigned to the category of threats to the system.

4) Subjective opinion and the impact exerted by the qualifications of specialized experts during compiling the list of threats are now practically disregarded.

As a result of the practical application of this method, it was revealed that the compiled list of threats exceeded the previous list by 17%, which was compiled by specialists before.

Models that simulate metagraph-based CN behavior open up the possibility of representing software components and all possible relationships among them. The software components are practical, system, and network software packages. All of these will lead to improved security for a particular system.

The developed method allows compiling a list of threats to the CS of any CNs and makes it possible to increase the number of detected threats. The approach can be extended to identify information security threats.

ACKNOWLEDGMENT

Selected findings of this work were obtained under the Grant Agreement in the form of subsidies from the federal budget of the Russian Federation for state support for the establishment and development of world-class scientific centers performing R&D on scientific and technological development priorities dated April 20, 2022, No. 075-15-2022-307.

REFERENCES

[1] Liu, Q., Zhang, T. (2023). Deep learning technology of computer network security detection based on artificial intelligence. Computers and Electrical Engineering, 110: 108813. https://doi.org/10.1016/j.compeleceng.2023.108813

[2] Racherache, B., Shirani, P., Soeanu, A., Debbabi, M. (2023). CPID: Insider threat detection using profiling and cyber-persona identification. Computers & Security, 132: 103350. https://doi.org/10.1016/j.cose.2023.103350

[3] Kuklin, V., Alexandrov, I., Polezhaev, D., Tatarkanov, A. (2023). Prospects for developing digital telecommunication complexes for storing and analyzing media data. Bulletin of Electrical Engineering and Informatics, 12(3): 1536-1549. https://doi.org/10.11591/eei.v12i3.4840

[4] Toch, E., Bettini, C., Shmueli, E., Radaelli, L., Lanzi, A., Riboni, D., Lepri, B. (2018). The privacy implications of cyber security systems. ACM Computing Surveys, 51(2): 1-27. https://doi.org/10.1145/3172869

[5] Liu, Y.C., Huang, C.M.K., Chang, Y.S., Lin, H.M., Chen, P.L. (2023). An integrative model of information processing and contextual factors on exploring information systems outsourcing success. Asia Pacific Management Review, 28(3): 327-335. https://doi.org/10.1016/j.apmrv.2022.12.001

[6] Gritzalis, S., Lian, S. (2013). Mathematical and computer modelling in information system security. Mathematical and Computer Modelling, 57(11-12): 2581-2582. https://doi.org/10.1016/j.mcm.2013.04.001

[7] Goel, S., Chen, V.Y. (2005). Information security risk analysis – A matrix-based approach. In Proceedings of the Information Resource Management Association (IRMA) International Conference, pp. 1-9.

[8] Dhillon, G., Smith, K., Dissanayaka, I. (2021). Information systems security research agenda: Exploring the gap between research and practice. The Journal of Strategic Information Systems, 30(4): 101693. https://doi.org/10.1016/j.jsis.2021.101693

[9] Ampatzoglou, A., Bibi, S., Avgeriou, P., Verbeek, M., Chatzigeorgiou, A. (2019). Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. Information and Software Technology, 106: 201-230. https://doi.org/10.1016/j.infsof.2018.10.006

[10] Abdrassilov, A., Orynbassarova, Y., Tvaronaviciene, M. (2023). Exploring environmental factors for the sports clusters development. Journal of Environmental Management and Tourism, 14: 799-810. https://doi.org/10.14505/jemt.v14.3(67).19

[11] Hettiarachchi, S., Wickramsinghe, S. (2016). Study to identify threats to information systems in organizations and possible countermeasures through policy decisions and awareness programs to ensure the information security. Information Security, 1-13.

[12] Nicho, M., Kamoun, F. (2014). Multiple case study approach to identify aggravating variables of insider threats in information systems. Communications of the Association for Information Systems, 35. https://doi.org/10.17705/1cais.03518

[13] Samson, G.L., Usman, M.M. (2015). Securing an information system from threats: A critical review. International Journal of Computer Applications Technology and Research, 4(6): 425-434. https://doi.org/10.7753/ijcatr0406.1002

[14] Hamid, B., Weber, D. (2018). Engineering secure systems: Models, patterns and empirical validation. Computers & Security, 77: 315-348. https://doi.org/10.1016/j.cose.2018.03.016

[15] Sony, M., Naik, S. (2020). Industry 4.0 integration with socio-technical systems theory: A systematic review and proposed theoretical model. Technology in Society, 61: 101248. https://doi.org/10.1016/j.techsoc.2020.101248

[16] Godquin, T., Barbier, M., Gaber, C., Grimault, J.L., Le Bars, J.M. (2020). Applied graph theory to security: A qualitative placement of security solutions within IoT networks. Journal of Information Security and Applications, 55: 102640. https://doi.org/10.1016/j.jisa.2020.102640

[17] Basu, A., Blanning, R.W. (2007). Metagraphs and Their Applications. Springer New York, NY. https://doi.org/10.1007/978-0-387-37234-1

[18] Wang, X.M., Li, Q., Yu, D., Wang, Z., Chen, H., Xu, G. (2022). MGPolicy: Meta graph enhanced off-policy learning for recommendations. Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, pp. 1369-1378. https://doi.org/10.1145/3477495.3532021

[19] Šajna, M., Wagner, A. (2023). Using edge cuts to find

Euler tours and Euler families in hypergraphs. Discrete Mathematics, 346(10): 113537. https://doi.org/10.1016/j.disc.2023.113537
[20] Hinding, N., Sugeng, K.A., Nurlindah, Wahyudi, T.J., Simanjuntak, R. (2022). Two types irregular labelling on dodecahedral modified generalization graph. Heliyon, 8(11): e11197. https://doi.org/10.1016/j.heliyon.2022.e11197
[21] Ping, P., Xuan, Z., Xinyue, M. (2017). Research on security test for application software based on SPN. Procedia Engineering, 174: 1140-1147. https://doi.org/10.1016/j.proeng.2017.01.267
[22] Kuklin, V.Z., Alexandrov, I.A., Umyskov, A.A., Lampezhev, A.K. (2022). Analysis of the prospects for developing storage and processing complexes for multiformat media data. Journal of Computer Science, 18(12): 1159-1169. https://doi.org/10.3844/jcssp.2022.1159.1169
[23] Shelupanov, A., Evsyutin, O., Konev, A., Kostyuchenko, E., Kruchinin, D., Nikiforov, D. (2019). Information security methods-modern research directions. Symmetry, 11(2): 150. https://doi.org/10.3390/sym11020150
[24] Stergiou, T., Leeson, M., Green, R. (2004). An alternative architectural framework to the OSI security model. Computers & Security, 23(2): 137-153. https://doi.org/10.1016/j.cose.2003.09.001
[25] Konev, A., Shelupanov, A., Kataev, M., Ageeva, V., Nabieva, A. (2022). A Survey on threat-modeling techniques: protected objects and classification of threats. Symmetry, 14(3): 549. https://doi.org/10.3390/sym14030549
[26] Federal Service for Technical and Export Control website, n.d. Available online: https://fstec.ru/.

NOMENCLATURE

OS	operating system
CN	computer network
IS	information system
CS	cybersecurity
IP	information protection