symmetry

MDPI

Article

Model of Threats to Computer Network Software

Aleksey Novokhrestov *, Anton Konev and Alexander Shelupanov

Department of Complex Information Security of Computer Systems, Tomsk State University of Control Systems and Radioelectronics, 40 Lenina Prospect, 634050 Tomsk, Russia; kaa1@keva.tusur.ru (A.K.); saa@keva.tusur.ru (A.S.)

* Correspondence: nak@fb.tusur.ru; Tel.: +7-(3822)-70-15-29

Received: 31 October 2019; Accepted: 10 December 2019; Published: 11 December 2019

Abstract: This article highlights the issue of identifying information security threats to computer networks. The aim of the study is to increase the number of identified threats. Firstly, it was carried out the analysis of computer network models used to identify threats, as well as in approaches to building computer network threat models. The shortcomings that need to be corrected are highlighted. On the basis of the mathematical apparatus of attributive metagraphs, a computer network model is developed that allows to describe the software components of computer networks and all possible connections between them. On the basis of elementary operations on metagraphs, a model of threats to the security of computer network software is developed, which allows compiling lists of threats to the integrity and confidentiality of computer network software. These lists include more threats in comparison with the considered analogues.

Keywords: computer network model; threat model; threat classification

1. Introduction

The problem of ensuring the security of computer networks has not lost its relevance from the moment of their appearance and wide distribution to the present day. Thus, according to a study by Positive Technologies, in 2018, as part of an external penetration testing, the network perimeter of 92 percent of companies was breached [1]. Along with this, technologies are constantly evolving. New types of threats appear [2], and the security of computer networks is evolving into the security of the Internet of Things [3–6].

An essential step in the process of providing security is to identify a list of relevant threats. However, before determining the relevance, it is necessary to compile the most extensive list of threats [7], in other words, to identify threats.

Network security issues are relevant for both large companies and small organizations [8]. At the same time, it is obvious that the resources that can be allocated for security will differ. This affects not only the possible costs of technical equipment, but also the qualifications of the specialists which the organization can hire. The professional level, as well as the subjective opinion of an expert when using existing approaches to building lists of threats to information systems, significantly affects the result.

An urgent task is to develop an effective methodology for compiling a list of threats to information security, the use of which will minimize the impact of the professional level and subjective opinion of an expert. This study is part of the development of a comprehensive approach to assessing the security of the information systems conducted in Tomsk University of Control Systems and Radioelectronics [9].

This paper addresses the issue of identifying security threats to computer network software. The aim of the study is to increase the number of identified threats. At the same time, issues of

determining the relevance of threats and further risk analysis remain outside the scope of this work. To achieve this goal it is necessary:

1. To analyze the current state of the subject area: computer network models and approaches to building threat models used in compiling lists of threats.
2. To develop a computer network model that allows to describe the structure of the system at a level of detail enough to compile a list of threats.
3. To develop a computer network threat model that takes into account the maximum possible number of threats.

With computer networks we mean local area networks, which are a system that provides data exchange between subnets, network nodes, and the software installed on them.

2. Related Work

There are many approaches to building a threat model. In [10] it is indicated that in threat modeling, there are techniques that center on attackers, assets, or software. It includes the STRIDE threat model [11], attack trees originally presented by B. Schneier, attack libraries, and privacy tools. In [12] authors deal with the threat classification problem and its motivation. They categorize threat classification approaches into two main classes: methods based on attacks techniques and methods based on threats impacts.

It should be clarified that the concepts of threat classification and threat modeling in the context of different works may differ. Classification is understood as a ride to gain an understanding of the characteristics and nature of known threats [12]. Threat modeling involves determining a list of threats to the security of the system or information used to further risk assessment and building a protection system [13].

Moreover, threat classification methods are used in threat modeling, which is justified. If there is a classification, it is easier for a specialist to navigate in the whole variety of existing threats. This approach to threat modeling is called high-level. On the other hand, using only classifications, it is difficult to obtain a detailed list of threats on the basis of which it is possible to build the structure of a protection system. Examples of such approaches can be considered in [11] and [14].

Low-level approaches are those that describe threats in detail. Such approaches may be based on the use of the list of attacks [15–19] or the list of attack scenarios [20]. Some approaches come down to analyzing the exploitation of vulnerabilities in the system [21–23].

In [24] a classification of threats is proposed that has signs of a high-level and low-level approach. The work is aimed at describing the threat's class impact instead of a threat impact as a threat varies over time. However, for its effective application in practice, there is not enough formalization.

The problem with many approaches is the lack of formalization, which leads to their ambiguous interpretation and subjectivity of the resulting list of threats. There are works that use the mathematical apparatus of graph theory, but they are aimed at formalizing the description of attacks, not threats themselves [25–27]. Some works are aimed at the description of attackers and does not allow to determine the list of threats [28].

Separately, it is necessary to mention the databases of threats, attacks, and vulnerabilities that are often used in practice when building threat models, such as the ATT&CK Matrix [29] and Information Security Threat Databank of FSTEC of Russia [30]. In connection with the specifics of the study [30], a detailed comparison of the results of the work was carried out with a list of threats mentioned in it.

In the analysis of approaches to building models of threats to the security of information systems and, in particular, computer networks, the following shortcomings were identified:

1. Some threat models contain elements of the attacker model, or the attacker model directly influences the formation of the list of threats.
2. In one threat model at one level there may be a generalized description of threats, as well as a description of special cases.

3. There is no division into threats aimed at the system and threats aimed at the information.
4. The building of threat lists is based on the subjective opinion of an information security specialist.

The key disadvantage of all models is that none of them explicitly describes threats to the information system. All attention is paid to the security threats to information processed in the information system.

Each of the considered models can take into account certain threats that are not described in another.

Furthermore, in many of the considered models there is no mathematical formalization, that is, threats are presented through verbal descriptions. The sequence of identification of threats to the system under consideration is given by general instructions, without a step-by-step description of the actions. This often leads to the fact that experts can interpret the same technique differently, moreover, experts often do not have a direct relationship with the organization, which introduces additional inaccuracies in the formation of a threat model.

Another drawback of existing approaches is the lack of justification for the classification of threats and consequently the lack of justification for the completeness of the proposed classification.

As a result of the analysis of approaches to building computer network models [31–35], we can conclude that, with their help, it is impossible to describe in detail what the objects in the information system are (that is, describe their parameters), as well as describe how they interact with each other. In order to more fully describe the threats to the information security of a computer network, the model of the computer network should satisfy the following requirements. It is necessary to take into account:

1. The hierarchy of computer network software.
2. The possibility of the existence of several connections between two elements.
3. The elements and the connections between them have parameters.

3. Proposed Approach

3.1. Computer Network Model

A computer network model based on attributive metagraphs allows to describe the software components of computer networks and all possible connections between them. The study considers only the software elements of computer networks (computer network software components and applications) and the links between them. The software in this case includes the application, system, and network software. A similar approach to the classification of system elements was applied in [36]. Links are implied not only between elements located at the same level, but also by indicating the nesting of one element in another. That is, application software operates in operating systems, which are system software. In turn, operating systems operate within the framework of local area networks (or subnets) implemented through network software. Thus, three levels of computer network software are distinguished. For convenience, the levels are designated as the application level, the operating system level, and the network level.

As a mathematical apparatus for the implementation of the model, attributive metagraphs were chosen [37]. The metagraph contains and coordinates among itself two main properties of the system: unity (a set of interlinked elements) and divisibility (each element of the system is also a system). In this regard, subsystems can be distinguished from the system. This allows to focus on the system or its subsystem if necessary.

The attributive metagraph nested at n levels of depth is represented as an ordered pair:

$$G = (X, E), \quad (1)$$

where G is the attributive metagraph nested at n levels of depth; $X = \{x_i\}, i = \overline{1,n}$ is nonempty finite set of vertices; $E = \{e_k\}, k = \overline{1,m}$ is nonempty finite set of edges.

Each edge of an n-dimensional graph connects two subsets of the 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 nesting level.

There are also functions that indicate the nesting of vertices and edges of a metagraph:

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

where $l, p, r, \dots, t$ is number of vertices and edges at the appropriate level.

The vertices and edges of the attribute metagraph are characterized by many attributes:

$$x_i = \{atr_j\}, \tag{4}$$

$$e_k = \{atr_h\}, \tag{5}$$

where x_i is a vertex of the metagraph, $x_i \in X$; e_i is an edge of the metagraph, $e_i \in E$; atr_j and atr_h are attributes of vertices and edges, respectively.

Thus, the elements of the computer network applications and the connections between the elements are represented by the following symmetric sets:

$X_1 = \{x_1^k\}$, $k = \overline{1,q}$ is a set of applications;
$X_2 = \{x_2^l\}$, $l = \overline{1,r}$ is a set of operating systems;
$X_3 = \{x_3^m\}$, $m = \overline{1,s}$ is a set of networks;
$E_1 = \{e_1^n\}$, $n = \overline{1,t}$ is a set of links between applications, defined over a set X_1;
$E_2 = \{e_2^o\}$, $o = \overline{1,u}$ is a set of links between operating systems, defined over a set X_2;
$E_3 = \{e_3^p\}$, $p = \overline{1,v}$ is a set of links between networks, defined over a set X_3.

The entire computer network can be represented as the following attributive metagraph, or the ordered sequence of six values:

$$G = (X_1, X_2, X_3, E_1, E_2, E_3). \tag{6}$$

Moreover, there are functions that indicate the occurrence of applications in operating systems and operating systems in networks:

$$f_1^w: g_1^w(x_1^k, e_1^n) \to x_2^l, \tag{7}$$

where x_1^k is an element of the set of applications; e_1^n is an element of the set of links between applications; x_2^l is an element of the set of operating systems.

$$f_2^y: g_2^y(x_2^l, e_2^o) \to x_3^m, \tag{8}$$

where x_2^l is an element of the set of operating systems; e_2^o is an element of the set of links between operating systems; x_3^m is an element of the set of networks.

The vertex is characterized by a set of attributes:

$$x_i^b = \{atr_a\}, \tag{9}$$

where $i = \overline{1,3}$ is the level of nesting of the vertex; b is the vertex number at a corresponding level i; atr_a are the attributes of the vertex (number, line, etc.).

The edge is characterized by a set of attributes:

$$e_i^h = \{x_i^s, x_i^e, \{atr_z\}\}, \tag{10}$$

where x_i^s, x_i^e are vertices connected by the edge; $i = \overline{1,3}$ is the level of nesting of the edge; atr_z are the attributes of the edge (number, line, etc.).

Table 1 shows the potential attributes of the elements of the sets in question.

In addition, a rule is introduced that a link between two elements at the *i*-th level exists if and only if a link exists between all elements located at higher levels to which objects of the i-th level belong. This means that applications installed on different operating systems are interlinked only if the corresponding operating systems are also interlinked. Similarly, operating systems in different networks can be interlinked only if such networks are interlinked as well.

Table 1. Attributes.

Elements	Attributes
Element of set X_1 (applications)	Application name Application version Number of the port used by the application
Element of set X_2 (operating systems)	OS name OS version IP-address used by the OS
Element of set X_3 (networks)	Network name Protocols in the network (OSI model network layer) Routing table IP-address and network mask
Element of set E_1 (links between applications)	OSI model application layer (session, presentation)
Element of set E_2 (links between operating systems)	Protocols of the OSI model transport layer
Element of set E_3 (links between networks)	Protocols of the OSI model network layer

With using the developed model, it is possible at the design stage of the system structure to take the characteristics of the elements and the relationships between them into account for requirements for the functions of information security tools.

The following is an example of using the model to describe a computer network. A small computer network consists of three computers, one of which has a mail server, and two mail clients. Since we consider only software, computers are represented by operating systems. To provide an example, not all communications between operating systems and software are provided. A graphical representation of the metagraph describing this network is presented in Figure 1.

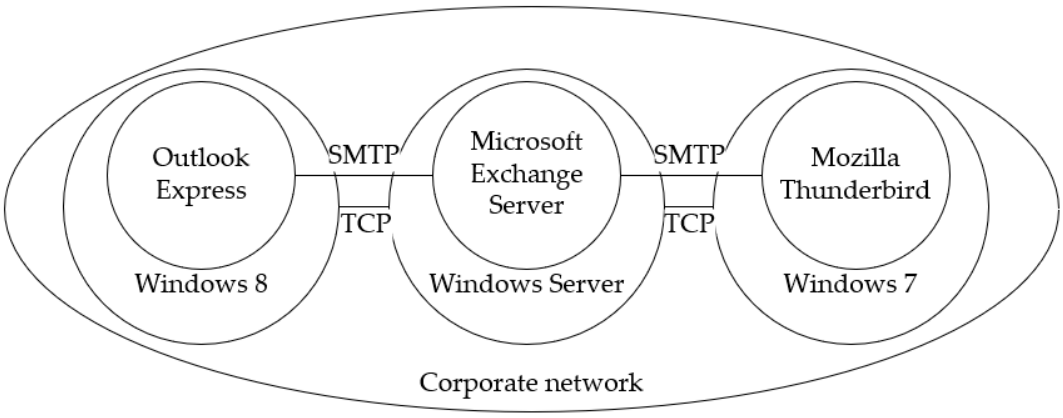

Figure 1. Metagraph representing the example network.

In terms of the proposed model, a computer network will be described as follows:

$$G = (X_1, X_2, X_3, E_1, E_2). \quad (11)$$

Set X_1 represents a list of software, set X_2 represents a list of operating systems and set X_3 a list of computer networks. Sets E_1 and E_2 contain lists of relationships between software and operating systems. Next, is an example:

$$X_1 = \{Outlook_1^1, Exchange_1^2, Thunderbird_1^3\}, \quad (12)$$

$$E_2 = \{TCP_2^1, TCP_2^2\}. \quad (13)$$

The functions indicating the nesting of software in operating systems will be as follows:

$$f_1^1 : g_1^1(Outlook_1^1) \to Windows8_2^1. \quad (14)$$

Other nesting functions looks similar.

3.2. Model of Threats

The proposed approach to the classification of threats and the developed threats model are based on elementary operations on metagraphs [37]. As shown earlier, a computer network is considered as a structure of interacting elements (vertices of the graph) and the links between them (edges of the graph). Threats are understood as an unauthorized change in the structure of a computer network (graph).

At this stage, it is necessary to indicate that the study considers only threats to the security of the system, not the information. At the same time, the classification of threats by violated properties is taken as the basis: confidentiality, integrity, and availability. The threats to the availability of the system are not considered, since when combining the lists of threats to the security of information and of the system, these threats will coincide. Thus, threats to the integrity and confidentiality of computer network software are considered.

The basic operations on attributive metagraphs include adding a vertex or an edge, removing a vertex or an edge and changing a vertex or edge attribute [38].

Based on this, the following classes of threats to the integrity of a computer network are proposed:

1. Threats of an element substitution—C_{s1X}
2. Threats of a link substitution—C_{s1E}
3. Threats of an element removal—C_{s2X}
4. Threats of a link removal—C_{s2E}
5. Threats of an element addition—C_{s3X}
6. Threats of a link addition—C_{s3E}
7. Threats of an element settings changing—C_{s4X}
8. Threats of a link settings changing—C_{s4E}

The threat of an element or link removal is characterized by the removal of a vertex or edge from the set X_i or E_j, respectively. Thus, for a set of applications, it is characterized as follows:

$$G' = (X_1 \setminus x_1^k, X_2, X_3, E_1, E_2, E_3), \quad (15)$$

where X_1 is a set of applications; x_1^k is a remote application and $x_1^k \in X_1$.

The threat of an element or link addition is characterized by the adding of a vertex or edge from the set X_i or E_j, respectively. Thus, for a set of applications, it is characterized as follows:

$$G' = \left(X_1 \cup x_1^{q+1}, X_2, X_3, E_1, E_2, E_3\right), \quad (16)$$

where x_1^{q+1} is an added application.

The threat of an element or link substitution is characterized by removing a vertex or an edge from the set X_i or E_j, respectively, and adding a vertex or an edge instead of the deleted one, i.e., for a set of applications, this is described by the sequence of equations (15) and (16):

$$G' = (X_1 \setminus x_1^k, X_2, X_3, E_1, E_2, E_3), \quad (17)$$

$$G'' = (X_1 \cup x_1^{k'}, X_2, X_3, E_1, E_2, E_3). \quad (18)$$

The threat of an element or link settings changing is carried out by changing an attribute of a vertex or an edge:

$$atr_a := atr_a'. \quad (19)$$

The following threat classes are proposed as a classification of computer network confidentiality threats:

1. Threats of an element name disclosure—K_{s1X}
2. Threats of a link name disclosure—K_{s1E}
3. Threats of an element settings disclosure—K_{s2X}
4. Threats of a link settings disclosure—K_{s2E}

In graph theory, the confidentiality threats of a computer network are described as the intersection of sets of protected elements, information about which should be hidden, with sets of well-known elements. Hence, the threat of disclosure (leakage) of information about the name of the application is characterized by the intersection of the set X_1 with the set J_1:

$$G' = (X_1 \cap J_1, X_2, X_3, E_1, E_2, E_3), \quad (20)$$

where $X_1 \cap J_1 = \{x_1^k | x_1^k \in X_1 \wedge x_1^k \in J_1\}$; x_1^k is an element belonging to the set X_1; X_1 is set of applications that needs to be protected; J_1 is set of applications whose elements are well-known.

The result of the study is a computer network threat model that integrates classes of threats K_S and C_S:

$$T_S = K_S \cup C_S, \quad (21)$$

where K_S is threats to the confidentiality of computer network elements; C_S is threats to the integrity of computer network elements.

At the same time, each of the 12 presented threat classes contains three types of threats: threats at the application layer, threats at the operating system layer, and threats at the network layer. In total, 36 types of threats to the information security of computer network software are obtained. The final threat classification is presented in Figure 2.

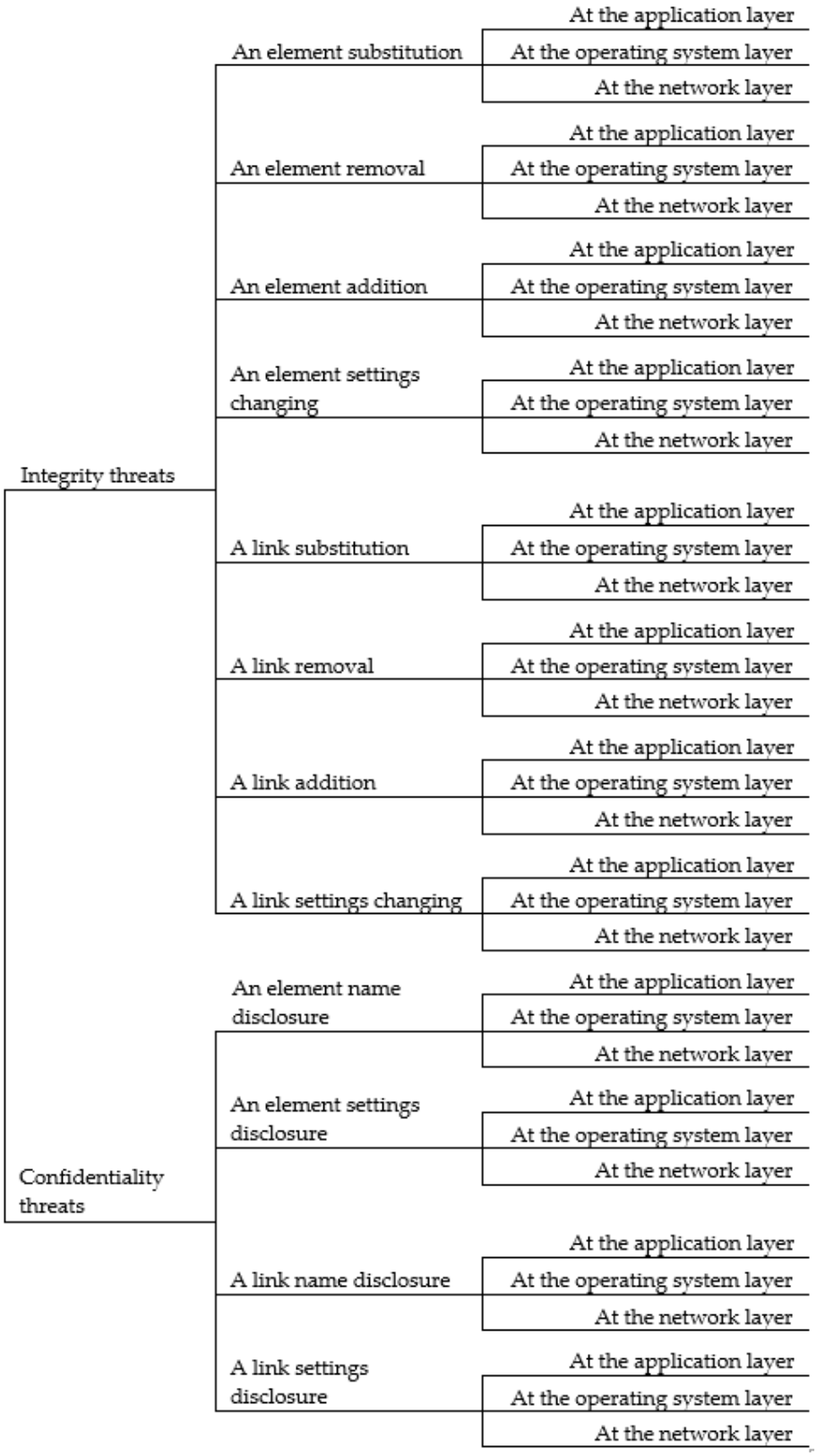

Figure 2. Proposed computer network software threats' classification.

On the basis of the use of basic operations on attribute metagraphs to determine the threat classes, it becomes possible to make an assumption about the completeness of the proposed classification of security threats to computer network software.

4. Discussion

The following is a comparison of the types of threats identified in this paper with those in [36], which uses a similar approach to the classification of system elements. In [36], four levels of system elements are distinguished: the physical layer, network layer, operating system (OS) layer, and application layer. The physical layer in connection with the limitations established in this paper is not considered in the comparison. The following threats are listed as threats to the software:

- Network layer: the availability of equipment is isolated, network traffic is intercepted, network traffic is modified.
- OS layer: malicious software is installed, the stability of system processes and services is violated, information resources are impacted.
- Application layer: applications are disabled, information resources of applications are impacted, the operations of applications are modified.

Some of these threats are explicitly threats to information, and therefore are not considered in the comparison. The result of the comparison is shown in Table 2. The intersections of the lines with the threat classes of the author's model with the columns in which the software levels of the computer network are indicated the identified types of threats. The marked cell means that in [36] security threats to the system related to this type were found. The unmarked cell means that no threats that could be attributed to this type were detected.

Table 2. Comparison of threats from [36] with the types of threats of the author's threat model.

Threat classes	Computer network software layers		
	Application	OS	Network
K_{s1X}			
K_{s1E}			
K_{s2X}			
K_{s2E}			
C_{s1X}	+	+	
C_{s1E}			
C_{s2X}	+		+
C_{s2E}			+
C_{s3X}	+		
C_{s3E}			
C_{s4X}			
C_{s4E}			

The information presented in the table shows that the proposed model describes a significantly larger number of types of threats than the considered counterpart. However, the approach in [36] allows the specialist to add other threats to the review, which makes the comparison incorrect. For a detailed comparison, we selected a list of threats from the Security Threat Databank of FSTEC of Russia [30].

In the course of the comparison, all 213 information security threats from [30] were classified by exposed object. Since the data bank defines threats as violating the confidentiality, integrity, and availability of information, it is difficult to identify threats to the information system among them. Threats were attributed to threats to the system in the case of a clear indication in the description of the threat that it violates any of the properties of the system. Threats in the description of which meant a violation of the properties of information due to gaining access to the system were considered as threats of information. As a result, 68 threats to the security of the information system were

identified. All these threats were correlated with the threat types identified during the development of the threat model.

The generalized comparison result is presented in Table 3. The marked cell in [30] means that security threats to the system related to this type were found. The unmarked cell means that no threats that could be attributed to this type were detected.

On the basis of the comparison results, it was found that the proposed approach to building a threat model allows information protection specialists to consider, when building an information protection system, 11 more types of information security threats than when using [30]. In total, according to the author's classification, 36 types of threats to the confidentiality and integrity of the system were identified, 25 of them were presented in [30].

Table 3. Comparison of threats from [30] with the types of threats of the author's threat model.

Threat classes	Computer network software layers		
	Application	OS	Network
K_{s1X}		+	
K_{s1E}		+	+
K_{s2X}	+	+	+
K_{s2E}		+	+
C_{s1X}	+	+	+
C_{s1E}	+		
C_{s2X}	+	+	+
C_{s2E}		+	
C_{s3X}	+	+	
C_{s3E}	+	+	
C_{s4X}	+	+	+
C_{s4E}	+	+	

One of the earliest versions of the approach proposed in this work was used to compile a list of threats to an automated system for commercial accounting of energy resources [39]. As a result, a list of 70 threats to the integrity of the system was compiled. Threats were considered at the software and hardware levels for the three main types of system elements and connections between them. The list obtained using the author's methodology and models turned out to be 18 percent more than that previously compiled by customer experts (59 threats to the integrity of the system).

It should be noted that formalization also has some disadvantages. Firstly, the complexity of formalized models can narrow the circle of people who can apply this model. Secondly, compiling threat lists using the developed formalized models may require a specialist to take a lot of time, especially for large computer networks that include dozens and hundreds of elements. However, both mentioned disadvantages will not matter if the proposed models are implemented in a software tool. The formalization of models allows to algorithmize the process of compiling lists of threats. Currently, the concept of a software tool is being developed. It is assumed that the specialist's task will be to compile a computer network model by specifying lists of elements and the relationships between them. Furthermore, a list of threats will be compiled automatically.

5. Conclusions

The analysis of the current state of the subject area—computer network models used to identify threats, as well as approaches to building computer network threat models—is carried out:

On the basis of the mathematical apparatus of attributive metagraphs, a computer network model was developed that allows to describe computer network software components (application, system, and network software) and all possible connections between them (network protocols, drivers, etc.).

Based on elementary operations on metagraphs, a model of threats to the security of computer network software was developed, which allows compiling complete lists of threats to the integrity and confidentiality of computer network software.

The relevance of threats is not considered in the framework of this work, however, it should be noted that the addition of one threat, for which it is necessary to introduce protection mechanisms, is already sufficient reason to consider an expanded list of threats.

Author Contributions: Conceptualization, A.K.; data curation, A.N. and A.K.; funding acquisition, A.S.; investigation, A.N.; methodology, A.S.; project administration, A.K.; supervision, A.S., writing—original draft preparation, A.N.; writing—review and editing, A.S. and A.K.

Funding: This research was funded by the Ministry of Education and Science of Russia, Government Order no. 2.8172.2017/8.9 (TUSUR).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design, execution, interpretation, or writing of the study.

References

1. Penetration Testing of Corporate Information Systems: Statistics and Findings, 2019. Available online: https://www.ptsecurity.com/ww-en/analytics/corp-vulnerabilities-2019 (accessed on 29 October 2019).
2. Internet Security Threat Report (ISTR) 2019. Symantec. Available online: https://www.symantec.com/security-center/threat-report (accessed on 29 October 2019).
3. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201, doi:10.1109/JIOT.2019.2935189.
4. Abdulghani, H.A.; Nijdam, N.A.; Collen, A.; Konstantas, D. A Study on Security and Privacy Guidelines, Countermeasures, Threats: IoT Data at Rest Perspective. *Symmetry* **2019**, *11*, 774, doi:10.3390/sym11060774.
5. Shelupanov, A.; Konev, A.; Kosachenko, T.; Dudkin, D. Threat model for IoT systems on the example of openUNB protocol. *Int. J. Emerg. Trends Eng. Res.* **2019**, *7*, 283–290, doi:10.30534/ijeter/2019/11792019.
6. Perera, C.; Barhamgi, M.; Bandara, A.; Ajmal, M.; Price, B.; Nuseibeh, B. Designing privacy-aware internet of things applications. *Inf. Sci.* **2019**, *512*, 238–257, doi:10.1016/j.ins.2019.09.061.
7. Konev, A.A. Approach to creation protected information model. *Proc. TUSUR Univ.* **2012**, *25*, 34–39. (In Russian)
8. Zahoor, A.S.; Mahmood, H.S.; Javed, A. Information security management needs more holistic approach: A literature review. *Int. J. Inf. Manag.* **2016**, *36*, 215–225, doi:10.1016/j.ijinfomgt.2015.11.009.
9. Shelupanov, A.; Evsyutin, O.; Konev, A.; Kostyuchenko, E.; Kruchinin, D.; Nikiforov, D. Information Security Methods—Modern Research Directions. *Symmetry* **2019**, *11*, 150, doi:10.3390/sym11020150.
10. Shostack, A. *Threat Modeling: Designing for Security*; John Wiley & Sons: Indianapolis, IN, USA, 2014; pp. 59–121.
11. The STRIDE Threat Model. Available online: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20) (accessed on 29 October 2019).
12. Jouini, M.; Rabai, L. Threat classification: State of art. In *Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security*; Gupta, B., Agrawal, D., Yamaguchi, S., Eds.; IGI Global: Hershey, PA, USA, 2016; pp. 368–392.
13. Wenjun, X.; Lagerström, R. Threat modeling—A systematic literature review. *Comput. Secur.* **2019**, *84*, 53–69, doi:10.1016/j.cose.2019.03.010.
14. Tang, J.; Wang, D.; Ming, L.; Li, X. A Scalable Architecture for Classifying Network Security Threats. Available online: http://papersub.academicpub.org/Global/DownloadService.aspx?ID=2514 (accessed on 29 October 2019).
15. Pan, J.; Zhuang, Y. PMCAP: A Threat Model of Process Memory Data on the Windows Operating System. *Secur. Commun. Netw.* **2017**, doi:10.1155/2017/4621587.
16. Ferrag, M.A.; Maglaras, L.A.; Janicke, H.; Jiang, J.; Shu, L. Authentication Protocols for Internet of Things: A Comprehensive Survey. *Secur. Commun. Netw.* **2017**, *2017*, doi:10.1155/2017/6562953.
17. Liu, F.; Li, T. A Clustering K-Anonymity Privacy-Preserving Method for Wearable IoT Devices. *Secur. Commun. Netw.* **2018**, *2018*, doi:10.1155/2018/4945152.

18. Wagner, T.D.; Palomar, E.; Mahbub, K.; Abdallah, A.E. Relevance Filtering for Shared Cyber Threat Intelligence (Short Paper). In *Information Security Practice and Experience*; Springer: Cham, Switzerland, 2017; pp. 576–586.
19. Lakhno, V. Creation of the adaptive cyber threat detection system on the basis of fuzzy feature clustering. *East. Eur. J. Enterp. Technol.* **2016**, *2*, 18–25, doi:10.15587/1729-4061.2016.66015.
20. Bodeau, D.J.; McCollum, C.D. *System-of-Systems Threat Model*; The Homeland Security Systems Engineering and Development Institute (HSSEDI) MITRE: Bedford, MA, USA, 2018.
21. Darwisha, S.; Nouretdinova, I.; Wolthusen, S.D. Towards Composable Threat Assessment for Medical IoT (MIoT). *Procedia Comput. Sci.* **2017**, *113*, 627–632, doi:10.1016/j.procs.2017.08.314.
22. Wu, Z.; Wei, Q. Quantitative Analysis of the Security of Software-Defined Network Controller Using Threat/Effort Model. *Math. Probl. Eng.* **2017**, *2017*, doi:10.1155/2017/8740217.
23. Azad, M.A.; Bag, S.; Perera, C.; Barhamgi, M.; Hao, F. Authentic-Caller: Self-enforcing Authentication in a Next Generation Network. *IEEE Trans. Ind. Inform.* **2019**, doi:10.1109/TII.2019.2941724.
24. Jouini, M.; Rabai, L.; Aissa, A.B. Classification of Security Threats in Information Systems. *Procedia Comput. Sci.* **2014**, *32*, 489–496, doi:10.1016/j.procs.2014.05.452.
25. Alhebaishi, N.; Wang, L.; Jajodia, S.; Singhal, A. Threat Modeling for Cloud Data Center Infrastructures. In *International Symposium on Foundations and Practice of Security*; Springer: Cham, Switzerland, 2016; pp. 302–319.
26. Johnson, P.; Vernotte, A.; Ekstedt, M.; Lagerström, R. pwnPr3d: An Attack-Graph-Driven Probabilistic Threat-Modeling Approach. In Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, Austria, 31 August–2 September 2016; pp. 278–283, doi:10.1109/ARES.2016.77.
27. Boukhtouta, A.; Mouheb, D.; Debbabi, M.; Alfandi, O.; Iqbal, F.; El Barachi, M. Graph-theoretic characterization of cyber-threat infrastructures. *Digit. Investig.* **2015**, *14*, S3–S15, doi:10.1016/j.diin.2015.05.002.
28. Luh, R.; Temper, M.; Tjoa, S.; Schrittwieser, S. APT RPG: Design of a Gamified Attacker/Defender Meta Model. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Madeira, Portugal, 22–24 January 2018; pp. 526–537.
29. MITRE ATT&CK Matrix. Available online: https://attack.mitre.org/ (accessed on 29 October 2019).
30. Information Security Threat Databank. Available online: https://bdu.fstec.ru/threat (accessed on 29 October 2019). (In Russian)
31. Bernard, G. Interconnection of Local Computer Networks: Modeling and Optimization Problems. *IEEE Trans. Softw. Eng.* **1983**, *9*, 463–470, doi:10.1109/TSE.1983.234782.
32. Dudin, E.B.; Smetanin, Yu, G. Problems and prospects of modeling computer information networks. A review. *Autom. Doc. Math. Linguist.* **2010**, *44*, 287–296.
33. Ansari, Y.E.; Myr, A.E.; Omari, L. Deterministic and Stochastic Study for an Infected Computer Network Model Powered by a System of Antivirus Programs. *Discret. Dyn. Nat. Soc.* **2017**, *2017*, doi:10.1155/2017/3540278.
34. Shchurov, A.A. A Multilayer Model of Computer Networks. *Int. J. Comput. Trends Technol.* **2015**, *26*, 12–16, doi:10.14445/22312803/IJCTT-V26P103.
35. Shchurov, A.A.; Marik, R. A Trusted Model of Complex Computer Networks. *J. ICT Stand.* **2016**, *3*, 201–230, doi:10.13052/jicts2245-800X.332.
36. Lavrova, D.S.; Pechenkin, A.I. Adaptive reflexivity threat protection. *Autom. Control Comput. Sci.* **2015**, *49*, 727–734, doi:10.3103/S0146411615080106.
37. Basu, A.; Blanning, R.W. *Metagraphs and Their Applications*; Springer: New York, NY, USA, 2007; pp. 53–64, doi:10.1007/978-0-387-37234-1.

38. Novokhrestov, A.; Konev, A. Mathematical model of threats to information systems. *AIP Conf. Proc.* **2016**, *1772*, 060015, doi:10.1063/1.4964595.
39. Novokhrestov, A.K.; Nikiforov, D.S.; Konev, A.A.; Shelupanov, A.A. Model of threats to automatic system for commercial accounting of power consumption. *Proc. TUSUR Univ.* **2016**, *19*, 111–114, doi:10.21293/1818-0442-2016-19-3-111-114. (In Russian)