JOURNAL OF APPLIED SCIENCE AND SOCIAL SCIENCE
eISSN 2229-3113 pISSN 2229-3205

**Volume 15 Issue 11, November 2025**
Impact factor: 2019: 4.679 2020: 5.015 2021: 5.436, 2022: 5.242, 2023: 6.995, 2024 7.75

# DEVELOPMENT OF SECURITY ANALYSIS ALGORITHMS IN THE ENTERPRISE CORPORATE NETWORK

**Sayfullayev Sherzod**
TUIT named after Muhammad al-Khwarizmi, senior-teacher
**Gulomov Sherzod**
TUIT named after Muhammad al-Khwarizmi, DSc
**Saidahmedov Mansur Smanalievich**
TUIT named after Muhammad al-Khwarizmi, Researcher

**Annotation.** This article suggests two main algorithms aimed at analyzing an enterprise's corporate network and increasing its security level – comparison (L-algorithm) and load-based algorithm. The comparison algorithm allows you to compare network objects based on their properties and break them down into security areas. On the other hand, the load-based algorithm quantifies the security characteristics of objects and groups them by vulnerability levels. Both algorithms are used in complex network security analysis and serve to increase the overall level of protection.

**Keywords:** network security, comparison algorithm, load-based algorithm, security domain, vulnerability analysis, corporate network.

The development of algorithms for analyzing the corporate network of an enterprise is inherently complex. As is known, each algorithm works with some initial data that must be provided in advance. In this case, the initial data is the structure of the enterprise network, or rather, the network section, the security of which must be assessed and, if possible, options for increasing the overall level of security of this section of the network. The first algorithm is called the comparative separation algorithm (L-algorithm), since it compares the types of services of network objects (G-algorithm) in a certain way. The second is a load-based algorithm, which is based on calculating the weights of network objects from a security point of view. The algorithms are implemented in different ways, but lead to the same result - the overall security of the network increases. Using one algorithm may not give accurate results. In this case, the second algorithm can be used. The application of algorithms is shown in the example at the end of the chapter.

When implementing the algorithm based on the distribution of security domains, the network structure is considered to be given according to the above criteria. That is, there is a graph G(U,V) of height n, where U is the set of network objects, V is the set of communication outputs connecting them. The number of features of each object of the network is determined, their number is the same for all objects and is equal to m. The vector of features of network objects $U=\bigcup_{i=1}^{n} u_i$ $\bar{t}=\{t_1,..., t_m\}$. It is natural that not every object has all the properties. If the property exists, the object is assigned a value of 1, otherwise it is assigned a value of 0. That is, Every object has a property. Here$u_i t_j^i = \begin{cases} 1 \\ 0 \end{cases}$ j=1,..., m, i=1,..., n.

Objects are grouped into security domains based on their security level. The security level S of an object is determined by the number and category of properties in this algorithm, or$u_i$

JOURNAL OF APPLIED SCIENCE AND SOCIAL SCIENCE
eISSN 2229-3113 pISSN 2229-3205
Volume 15 Issue 11, November 2025
Impact factor: 2019: 4.679 2020: 5.015 2021: 5.436, 2022: 5.242, 2023: 6.995, 2024 7.75

$S(u_i) = \bigcup_{j=1}^{m} t_j^i$, here .i=1,..., n

Thus, each object in the network has its own security level. To generalize objects into domains, the following is required. Objects are placed in a matrix of properties [1]:

$$\begin{pmatrix} t_1^1 & t_2^1 & \cdot & \cdot & t_m^1 \\ t_1^2 & t_2^2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ t_1^n & t_2^n & \cdot & \cdot & t_m^n \end{pmatrix}.$$

The $t_j$ rows of the matrix represent network objects, and the columns represent object properties. The categories of object properties are considered the same in one column. If an object has no property, then this property is equal to 0.

All objects are on the same network (logically). This allows you to avoid being tied to specific connections between objects.

A constraint is introduced, meaning that the columns of the matrix are not allowed to be rearranged. These conditions are useful when comparing rows. sort column elements by roware compared and the matrix rows are arranged in such a way that rows with the same content are close together. The primary distribution of network objects by security level is as follows [2]:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Several options are considered. Several rows remain equal (rows 3 and 4). This is the simplest case. In conclusion, network objects with this property clearly fall into one security domain.

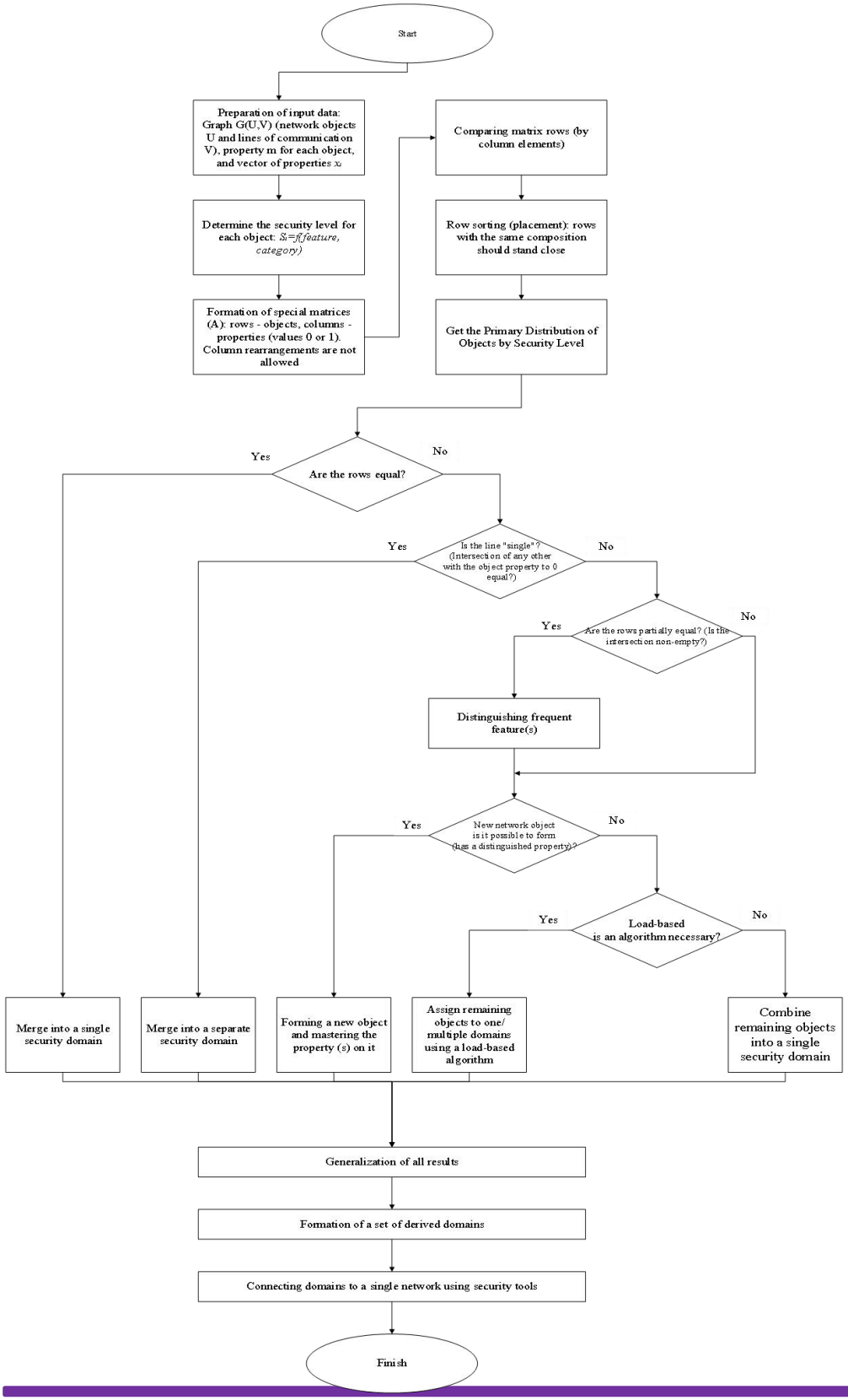
The intersection of these objects' properties with any other object's properties is 0. In other words, this object is considered unique. This falls into a separate security domain.

The set of rows is partially equal (rows 2-6). This means that the intersection of the set of object properties is not empty. In this case, it is necessary to isolate one or more properties that are frequently found in the objects under consideration (for example, a feature $t_2^i$). In this case, 2 options can be considered:

JOURNAL OF APPLIED SCIENCE AND SOCIAL SCIENCE
eISSN 2229-3113 pISSN 2229-3205

**Volume 15 Issue 11, November 2025**

Impact factor: 2019: 4.679 2020: 5.015 2021: 5.436, 2022: 5.242, 2023: 6.995, 2024 7.75



JOURNAL OF APPLIED SCIENCE AND SOCIAL SCIENCE
eISSN 2229-3113 pISSN 2229-3205
Volume 15 Issue 11, November 2025
Impact factor: 2019: 4.679 2020: 5.015 2021: 5.436, 2022: 5.242, 2023: 6.995, 2024 7.75

Figure 1. Security domain distribution comparison algorithm

- forming a new network object (one or more) and assimilating this feature (one or more) into it;
- if the previous steps are not possible, the remaining objects should be combined into a single security domain or separated into one (or more) domains. It is advisable to use the load-based algorithm described below for separation.

Thus, a set of security domains or subnets is created, since security domains form a specific defined subnet. The resulting domains must be connected to a single network. This requires the use of security tools between domains.

The mechanism of implementation of the algorithm, despite its simplicity, is effective. A negative effect can be attributed to the increase in the number of network objects, which is reflected in the cost. There are cases where this algorithm cannot be applied to the current network (for example, if each object is unique in terms of its set of properties). Therefore, the need arises to use a load-based algorithm, despite its complexity.

Load-based algorithm for security domain allocation. The initial data for this algorithm is the same as for the comparison algorithm based on the distribution of security domains. That is, there is a graph G(U,V) of height n, $U= \bigcup_{i=1}^{n} u_i$ where U is the set of network objects, and V is the set of links connecting them. The number of features of each network object is specified, the number of which is the same for all objects and is equal to m.

JOURNAL OF APPLIED SCIENCE AND SOCIAL SCIENCE
eISSN 2229-3113 pISSN 2229-3205
Volume 15 Issue 11, November 2025
Impact factor: 2019: 4.679 2020: 5.015 2021: 5.436, 2022: 5.242, 2023: 6.995, 2024 7.75

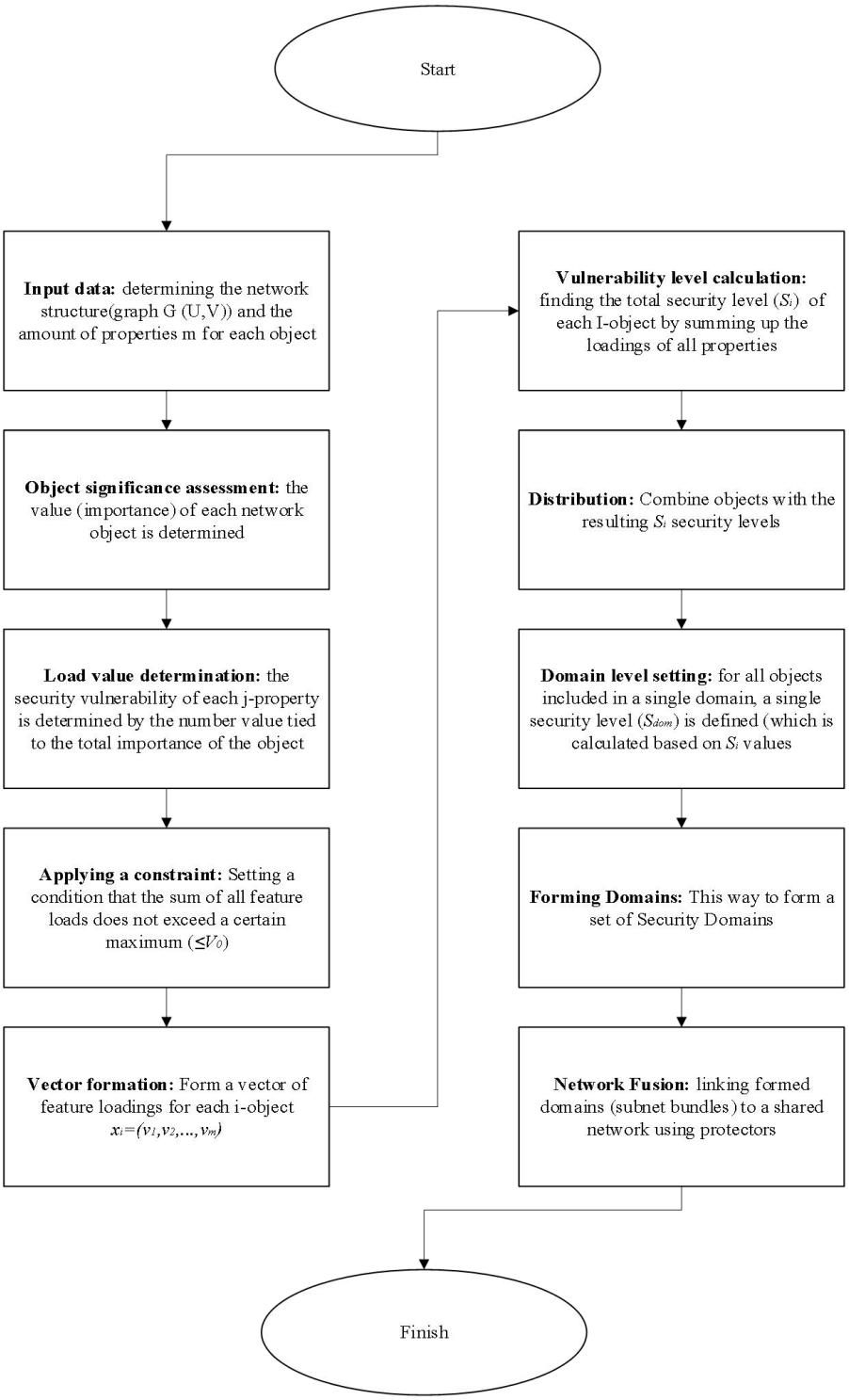



Figure 2. Load-based security domain allocation algorithm

It is known that each feature has a security vulnerability. Therefore, features differ from each other - one is more secure, the other is not. Each feature has a load, of course, security (that is

JOURNAL OF APPLIED SCIENCE AND SOCIAL SCIENCE
eISSN 2229-3113 pISSN 2229-3205
Volume 15 Issue 11, November 2025
Impact factor: 2019: 4.679 2020: 5.015 2021: 5.436, 2022: 5.242, 2023: 6.995, 2024 7.75

why it is called that). The load of a particular feature (or security) is determined based on the value (importance) of the network object. The network object feature vector $\bar{t}=\{t_1,..., t_m\}$ is defined as. The property indicates its loading numerical value. That is,$t_i$

$t_1=\begin{cases} p, \text{object feature load} \\ 0, \text{feature not available} \end{cases}$, here $p\in N$.

The loading p for the most important $t_k$ feature is chosen such that the sum of the feature loadings is less $t_1,..., t_{k-1}, t_{k+1},..., t_m$ than or equal to $t_k$:

$$t_k \leq \sum_{i\neq k} t_i \text{ , } i=1,..., m.$$

Thus, for each object, a set of features is obtained, where each feature is given a numerical value of its vulnerability, which indicates the importance of the network or object. This is achieved by calculating the sum of simple features [3]:

$S(u_i)=\sum_{j=1}^{m} t_j^i$, where $t_j^i$ is the j-property of the object $u_i$.

Objects are grouped into domains based on their security levels. Objects that fall into one domain receive the same or similar security levels $\delta$. The index $\delta$ is determined based on the security level values of the network objects.

The result of the above steps is a set of security domains, which are combined into a single network using a protection mechanism between domains.

This algorithm has several shortcomings. First, determining the loadings of the features is a complex task. Second, searching for the indicator $\delta$ is also complex. Simplification can be achieved by interacting with the user.

This algorithm is more complex than the comparison algorithm based on the distribution of security domains, but because of the different approach here, it can help when the comparison algorithm is not available.

In conclusion, the application of benchmarking and load-based algorithms to network architecture has shown that both algorithms are simple and effective. It has been found that the application of algorithms serves to increase overall network security.

**REFERENCES:**

1. Kubarenko A.S. Modernization of the enterprise's corporate computer network // Concept. - 2016. - T. 11. – S. 3131-3135.
2. Stolyarenko, A. V. Primenenie informatsionno-kommunicatsionnyx tekhnologii v deyatelnosti predpriyatiy turistskoy sfery / A.V. Stolyarenko, A.A. Danilchenko // Sovremennye nauchnye issledovaniya i innovatsii. – 2017. – No. 1.
3. Mukhin D.E., Stolyarenko A.V. Informatsionno-kommunicatsionnaya sreda turistskorekreatsionnogo predpriyatiya Kryma // Tavricheskiy nauchnyi obozrevatel. – 2017. – No. 1 (18).