UDC 519.715

# System construction of cybersecurity vulnerabilities with Q-analysis

V. I. Polutsyganova[1]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Educational and Research Institute of Physics and Technology*

**Abstract**

Today, in order to assess potential cyber threats, it is necessary to conduct a comprehensive assessment of the vulnerabilities of the investigated system. To do this, it is necessary to describe the identified vulnerabilities and consider potential vulnerabilities. In addition, the relationship between system vulnerabilities must be properly assessed. The most common assumption is that all vulnerabilities are independent and are implemented either by random events or by malicious intent. The paper proposes a method that allows modeling the vulnerabilities of complex systems as a whole, taking into account their hidden connections. Q-analysis [2] was used to study the structure of the system of interconnected vulnerabilities that arise in the process of project implementation. An example of the application of Q-analysis methods is presented and an explanation of the nature and impact of some potential threats and their combinations is offered.

*Keywords*: Q-analysis, simplicial complex, cyber systems, vulnerabilities

## Introduction

In order to assess all vulnerabilities in a cyber system, it is necessary to understand what vulnerabilities may or are actually occurring. Even more important is understanding how certain vulnerabilities directly or indirectly affect other vulnerabilities and even create new ones [1].

In order to identify them, this work describes the system of vulnerabilities using Q-analysis. This method is described by R. Atkin in [2], and its essence is that there are not only connections of a binary nature, but also connections of an n-ary nature. Uses the simplicial complex [3] instead of graphs to describe structures. Below are the types of simplicial complexes and different degrees of connectivity (Fig. 1).

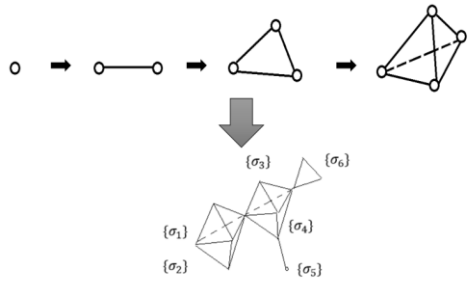


**Figure 1:** Simplicial complex

The description of the vulnerability system through the simplicial complex makes it possible to apply the Q-analysis apparatus. One of the most successful methods of representation are the so-called q-vectors or structural trees [3, 7], which provide an estimate of the complexity of the connections between vulnerable sites. Such a demonstration can reveal hidden connections and correctly calculate the risk to the system. An interesting task is the problem of constructing simplicial complexes for specific vulnerable systems.

The method of construction and restoration of the complex is described in [4]. Most importantly, it is necessary to use statistical data to assess the impact of certain vulnerabilities on others, as well as to create appropriate systems and integrate them into a complex. In more detail, this process will be described in an example.

## 1. Methodology for construction a vulnerability system

The methodology consists of several stages. The first of them is an assessment of possible and real vulnerabilities in the system. For this, you need to conduct a cybersecurity audit, there are enough tools for this. But, unfortunately, the output data of such tools gives an idea only that the detected vulnerabilities exist, while the

relationship between them is not evaluated in any way. Such clarification will make it possible to warn of the possibility of real threats and attacks, and to apply better cyber protection methods.

So, after conducting an audit and identifying a set of vulnerabilities, at the second stage it is necessary to build a matrix of incidence between vulnerabilities and those parts of the system that can potentially be affected by exploit attacks. To do this, it is necessary to identify the relevant vulnerabilities, compare them with the vulnerability database and determine which types of subsystems they can compromise. This means that it is necessary to construct a matrix V, dimensions mxn, m - the number of subsystems, n - the number of vulnerabilities for the entire system. Next, put one at the intersection of the name of the subsystem and the vulnerability, if this subsystem includes this vulnerability. In this way, we have a reflection of what specific vulnerabilities can occur in each subsystem, and therefore affect the system's operation.

The third stage should be the construction of an incidence matrix for subsystems. We build a square matrix *S*, dimensions, *mxm*, which displays connections for subsystems. This step is necessary when the vulnerabilities are not known to be related to each other. Then comparing the incidence matrices between vulnerabilities and subsystems, subsystems among themselves, we get the incidence matrix between vulnerabilities. It will display certain cascading dependencies and make it possible to determine the most significant vulnerabilities in the system. The required matrix is calculated as follows:

$I = V^T SV$.

We get an *nxn* matrix that reflects the relationships of vulnerabilities for this system.

The next stage will be the construction of a simplicial complex based on the relationship between vulnerabilities.

A simple complex is a multidimensional topological structure that better describes the relationship between the parts of a system and its elements than a graph. The term q-connection is also used in this context i.e. the level of connectivity between the simple in the complex. At any level of such a connection a simple complex can be defined as chains i.e. a graph whose nodes are simple and whose edge links are of smaller dimension than the degree of the given link.

Therefore, if two simplexes have $q$+1 common vertex (shared q-dimensional simplexes), then they are $q$-connected. The algorithm for finding q values for common faces of all pairs of simplexes in $K$ and the algorithm for obtaining $Q_q$ values uses the incidence matrix I, which defines $K$ [5].

The algorithm described in [5] is used to build the simplex complex. The point is that you need to go from a graph to a simplex complex.

To find the q-common faces of all pairs of Y-simplexes in K(X; λ), it is necessary to calculate:

• matrix $\Delta\Delta^T$ size (m×m);

• the difference $\Delta\Delta^T - \Omega$ , where Ω is a matrix consisting of units.

The integers on the diagonal of the obtained matrix are the dimensions of the simplexes Y. Such a representation will make it possible to evaluate not only the connection in the system of vulnerabilities, but also the multiplicity of such connections [5].

The assumption is used that the vulnerabilities of each of the subsystems can affect the emergence of other vulnerabilities, as a result of the interconnectedness of the subsystems. The scheme of the algorithm is shown in Fig. 2.

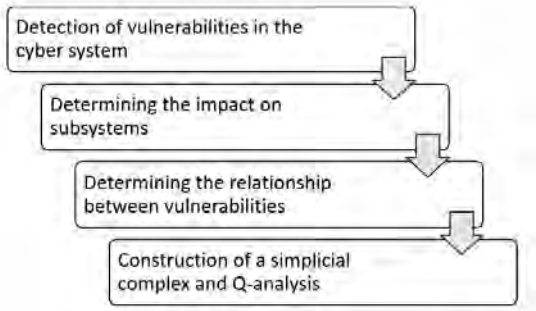


**Figure 2:** Scheme of the algorithm for construction a vulnerability system

Generalizing connections approach makes it possible to move from vulnerabilities to risk assessment, which will make it possible to prevent threats and attacks. So, this methodology has more practical significance.

## 2. Example of the vulnerability system construction for generalized cyber systems

First of all, to build a vulnerability system, you need to understand its structure and the technical base it uses. More often than not, such information is confidential or available only to a narrow circle of third parties who cooperate with the organization that owns the cyber system. Therefore, general information will be used in

this study. That is, the statistics of companies that deal with the detection of vulnerabilities. Based on statistics, we can understand which vulnerabilities occur more often and how they can affect each other. Using the methodology described above, we will build a general example for the system of vulnerabilities.

The Edgescan report [6] was used. Below are the statistics of the most frequently occurring vulnerabilities Tab.1.

**Table 1**
Statistics of the most frequently detected vulnerabilities

| №№ | Name | % of discovered Vulnerabilities |
|---|---|---|
| **V1** | Cross-Site Scripting - XSS (reflected) | 49.8% |
| **V2** | Broken Authentication/Poor Session Management, Brute Forcing Possible | 22.1% |
| **V3** | File path traversal/Information disclosure/Source Code Disclosure | 6.9% |
| **V4** | Authorisation Issue – Privilege Escalation | 6.0% |
| **V5** | File path traversal/Direct Object Access | 5.1% |
| **V6** | Malicious File Upload | 3.2% |
| **V7** | Deserialization Attacks | 3.2% |
| **V8** | Executable Code injection | 2.8% |
| **V9** | XML External Entity Injection (XXE) | 2.3% |
| **V10** | Server-Side Request Forgery (SSRF) | 1.8% |

Using additional information from the report, we will build an incidence matrix between vulnerabilities. For simplicity, let's number each vulnerability by $V_i$ (Tab.2).

**Table 2**
Incidence matrix between vulnerabilities

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| V1 | | | 1 | | 1 | 1 | | | | 1 |
| V2 | 1 | 1 | 1 | 1 | 1 | 1 | | | 1 | 1 |
| V3 | | | | 1 | | 1 | | 1 | | |
| V4 | | 1 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 |
| V5 | | 1 | | 1 | | 1 | | 1 | | |
| V6 | | | 1 | 1 | | 1 | 1 | 1 | | 1 |
| V7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| V8 | 1 | 1 | | 1 | 1 | 1 | 1 | 1 | | 1 |
| V9 | 1 | 1 | | 1 | | 1 | 1 | 1 | | |
| V10 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |

Next, using the algorithm for constructing the connectivity matrix for the simplicial complex, we use the algorithm described in [5]. We get data for the construction of the simplicial complex in Tab.3.

**Table 3**
Incidence matrix between vulnerabilities

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
|---|---|---|---|---|---|---|---|---|---|---|
| V1 | 3 | 4 | 1 | 4 | 3 | 3 | 4 | 4 | 3 | 4 |
| V2 | 4 | 7 | 3 | 7 | 5 | 5 | 8 | 8 | 6 | 6 |
| V3 | 1 | 3 | 3 | 4 | 3 | 4 | 4 | 4 | 4 | 3 |
| V4 | 4 | 7 | 4 | 8 | 5 | 7 | 9 | 8 | 6 | 8 |
| V5 | 3 | 5 | 3 | 5 | 4 | 4 | 5 | 5 | 5 | 4 |
| V6 | 3 | 5 | 4 | 7 | 4 | 6 | 7 | 6 | 5 | 6 |
| V7 | 4 | 8 | 4 | 9 | 5 | 7 | 9 | 9 | 7 | 8 |
| V8 | 4 | 8 | 4 | 8 | 5 | 6 | 9 | 8 | 7 | 7 |
| V9 | 3 | 6 | 4 | 6 | 5 | 5 | 7 | 7 | 6 | 5 |
| V10 | 4 | 6 | 3 | 8 | 4 | 6 | 8 | 7 | 5 | 7 |

As can be seen from the matrix, there are connections of the ninth dimension in the simplicial complex. This indicates that some vulnerabilities have a high level of impact on cyber security. Unfortunately, a complex of this dimension cannot be drawn in a space of dimension 3. But to track the impact of certain vulnerabilities, we will build a structural tree (Fig.4) using the algorithm given in [5].

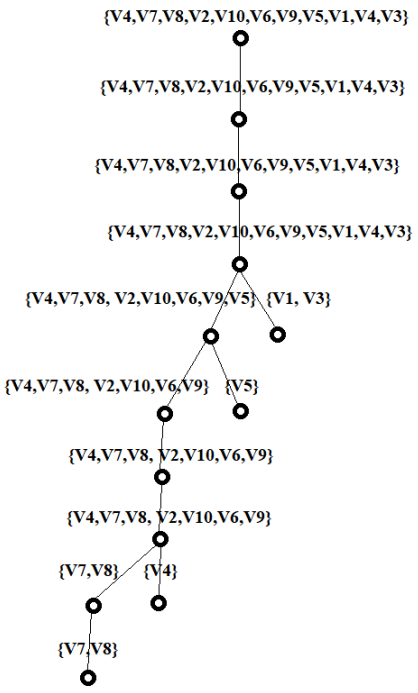


**Figure 4**: Structural tree

The structure tree shows how strongly interconnected vulnerabilities are. Each level highlights the degree of influence of each of the elements. As can be seen from the example, all vulnerabilities are to one degree or another interconnected by a 5-fold connection. This is a strong level of q-connectivity [5]. At the sixth level, the connection between $V_1$, $V_3$ and other vulnerabilities is lost and the simplicial complex of the system disintegrates into two simplexes. Given that the simplex itself has a corresponding vulnerability, there is no further influence from them. At the next level, $V_5$ stands out. At levels 7 and 8, the simplex of vulnerabilities is unchanged, indicating that this set of vulnerabilities is highly interconnected and mutually influencing. At the 9th level, a large number of vulnerabilities that were part of the simplex disappear, and it also splits into two more simplexes. At the last level, the smallest simplex with the largest dimension remained. It consists of two vulnerabilities that have a significant impact on the functioning of the generalized cyber system. Looking at the description of the corresponding $V_7$ - Deserialization Attacks, $V_8$ - Executable Code injection, we understand that such vulnerabilities do not occur often, but if they are present in the system, they entail the emergence of other vulnerabilities. This in turn calls into question the security and reliability of the corresponding cyber system.

Based on the conducted analysis, we conclude that, first of all, in the created or the created system, it is necessary to conduct testing to find these two vulnerabilities, because the potential consequences can be critical.

## Conclusions

In the course of the work, the methodology of building a system of cyber system vulnerabilities was presented. The main stages of the procedure for identifying and analyzing the impact of vulnerabilities between themselves and the system as a whole are given. In the considered example, the algorithm for building a simplicial complex and a structural tree is analyzed, followed by an analysis of the relationship between vulnerabilities. This work shows that some vulnerabilities that are often not detected in the system can have a large and indirect impact on the security, reliability and integrity of the system.

## References


[1] Качинський А. Б. Безпека складних систем / під ред. члена-кореспондента НАН України Довгого С. О.—К. : Юстон, 2017.—498 с.

[2] Atkin R. H. “Mathematical structure in human affairs”, Heinemann Educational Books, (1973); 143. doi: 10.1137/1018064.

[3] Beaumont J.R., Gatrell A.C. “An introduction to Q-analysis” Catmog 34, 1982. URL: https://alexsingleton.files.wordpress.com/2014/09/34-an-introduction-to-q-analysis.pdf.

[4] Polutsyganova V. I., Smirnov S. A. The inverse problem of Q-analysis of complex systems structure in cyber security / Scientific journal “Theoretical and Applied Cybersecurity” Vol. 4 No. 1 (2022) p. 61–68.

[5] Полуциганова В. І., Смирнов С. А. Методологія побудови основних метрик Q-аналізу та їх застосування/ / Системні дослідження та інформаційні технології. - 2019. - № 3. - С.76-88.

[6] Vulnerability Statistics Report 2023. Edgescan. URL: https://www.edgescan.com/intel-hub/stats-report/ (date of access: 22.06.2023).

[7] Jeffrey H. J. “Some structures and notation of Q-analysis”, Environment and Planning B Planning and Design, (1981). doi: 10.1068/b080073.