Hindawi
Security and Communication Networks
Volume 2019, Article ID 7510809, 8 pages
https://doi.org/10.1155/2019/7510809

Research Article

Sensitivity of Importance Metrics for Critical Digital Services Graph to Service Operators' Self-Assessment Errors

Mariusz Kamola

R&D Department, NASK National Research Institute, Kolska 12, 01-045 Warszawa, Poland

Correspondence should be addressed to Mariusz Kamola; mariusz.kamola@nask.pl

Received 22 March 2019; Accepted 31 August 2019; Published 23 September 2019

Academic Editor: Clemente Galdi

Interdependency of critical digital services can be modeled in the form of a graph with exactly known structure but with edge weights subject to estimation errors. We use standard and custom centrality indexes to measure each service vulnerability. Vulnerability of all nodes in the graph gets aggregated in a number of ways into a single network vulnerability index for services whose operation is critical for the state. This study compares sensitivity of various centralities combined with various aggregation methods to errors in edge weights reported by service operators. We find that many of those combinations are quite robust and can be used interchangeably to reflect various perceptions of network vulnerability. We use graphs of source files' dependencies for a number of open-source projects, as a good analogy for real critical services graph, which will remain confidential.

1. Introduction

Correct operation of digital services and infrastructures has since long become critical for societies, and therefore demands coordinated actions for maintenance and incident response. The Directive on Security of Network and Information Systems (NIS [1]), by the European Parliament, provides a framework for coherent implementation of security measures by European Union member states. Due to the scale and dynamics of digital networks, effective and efficient protection of their operation must be assisted by intelligent decision support systems operating at national level. Such systems should be

(i) Complete, i.e., possessing information about all critical services in the country

(ii) Automated, i.e., minimizing human factor in daily operations as well as in network model construction

(iii) Coupled, i.e., exchanging information at international level

Researchers, industry, and regulators stay aware of the above challenges and come up accordingly with ideas of such systems (cf., e.g., [2, 3] and references therein). Notably, Polish government is supporting National Cybersecurity Platform (NPC), a R&D project whose goal is to address the first two of the above issues, i.e., actually implement and deploy a system supporting security operation centers (SOCs). A crucial phase of NPC operation is creation of a graph modeling interdependent digital services run by various operators. This process is done semiautomatically from SOC perspective: service dependencies are discovered in depth-first search fashion, by interviewing subsequent operators with online questionnaires.

Apart from privacy and organizational obstacles, filling a questionnaire can be a challenge of its own for an operator. For a given own service, an operator is asked to report services preconditioning its correct operation, and to provide estimates of their impact on own service in terms of availability, confidentiality, and integrity (CIA) [4]. While the earlier is quite straightforward (as it can be based on inspection of business contracts, service level agreements (SLAs), and invoices or any other formal documents), measuring the magnitude of service dependencies is prone to errors and bias. But, on the other hand, the national critical services network model is built exactly with this info. The model includes routines for vulnerability calculation for each service. Vulnerabilities in turn get combined into a scalar index of overall network vulnerability.

Our goal is to examine how the above process is sensitive to incorrect information about mutual service impact as reported by operators, with the assumption that the structure of the network is known fully and correctly. Such information is crucial because that the scalar index value will be reported to SOCs and, consequently, will play the role of the main threat indicator.

We organized the paper as follows. A network model of services is presented in the remaining part of this section. A suite of methods for calculation of service vulnerability and for aggregation of vulnerabilities into a scalar vulnerability index are described in Section 2. It is followed by the section with discussion of results (Section 3), and we conclude in Section 4.

The network of interdependent digital services is modeled as a directed graph:

$$G(V, E), \tag{1}$$

where V is a list of ordered vertices, representing services, $V = (v_1, v_2, \ldots, v_{|V|})$, and E is a list of ordered edges: $e_{ij} \in E$ if operation of service v_i influences operation of service v_j. The impact of such influence is defined by the operator of service v_j on a discrete scale from 1 to 10. All the information about the graph structure and service impact can be expressed conveniently by adjacency matrix $\mathbf{A}$, whose element a_{ij} is equal to the impact value or zero if there is no edge e_{ij}. Here, we assume to operate with respect to only one impact aspect, e.g., how much the loss of service i availability influences service j availability. There can be nine such aspects in total, $\{C, I, A\} \times \{C, I, A\}$. It is possible to combine them all into one scalar coefficient, when some assumptions on their meaning are made, e.g., if one considers them as probabilities.

Such graph model extension with edge weights represented actually by a matrix of up to nine aspects of impact demands developing new graph algorithms—or picking up one of the aspects, like it is done in this paper. It makes the model universal enough to accommodate both digital services and physical infrastructure elements. In the latter case, one refers to just the availability aspect. For example, availability of backup power supply may influence availability and integrity of the physical access control system; hence, an operator has to address the influence in two aspects: $A \longmapsto A$ and $A \longmapsto I$.

Topology of a service graph represents existence of service interdependencies, while edge weights stand for intensity of those interdependencies. When combined, they make it possible to calculate the overall vulnerability of each service. There are many ways such vulnerability could be formulated; we express its definition as

$$\mathbf{r} \coloneqq \Phi(\mathbf{A}), \tag{2}$$

where Φ is some function defined over adjacency matrix that computes vector $\mathbf{r}$ of vulnerabilities for each service, respectively.

While $\mathbf{r}$ contains complete information about vulnerability of each service, a single scalar index γ of overall network vulnerability would be much more convenient in everyday use. Like for individual vulnerabilities, its calculation can be accomplished in many ways; we denote this process as

$$\gamma \coloneqq \Gamma(\mathbf{r}), \tag{3}$$

where Γ is some function defined over vulnerability vector.

The major practical problem concerns credibility of γ, which is computed indirectly from $\mathbf{A}$ whose values are not objective. They come from the questionnaires and are a result of self-assessment process by service operators, whose accuracy depends on their cybersecurity awareness and maturity of methodologies used in service impact estimation. An objective approach to vulnerability estimation would require excessive provocative tests on critical services or postmortem analyses, both of which are costly and undesirable.

Therefore, we must assume that, contrary to structure of service dependencies that is known and correct, the reported impact values $\widetilde{A}$ differ from true ones by some errors:

$$\widetilde{a}_{ij} = \begin{cases} \min\left(10, \max\left(1, a_{ij} + \xi\right)\right), & \text{if } a_{ij} > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where ξ is realization of a random variable with uniform discrete distribution $\mathcal{U}\{-N, N\}$. Here, N is the maximum impact estimation error in the ten-star rating scale. Note that in (4), we curb disturbed rating within the original scale of one to ten stars. Consequently, we denote calculated vulnerabilities of serivces for the reported values of $\widetilde{A}$ as

$$\widetilde{r} \coloneqq \Phi(\widetilde{A}). \tag{5}$$

Star ratings have been commonplace practice in many fields where user feedback is required. While facilitating the questioning process from a psychological perspective, it complicates analysis of statistical properties of responses, as it has been reported in [5]. The same authors claim that scales with more than seven stars provide too many possibilities and spoil the quality of a poll. Likewise, providing the respondent a scale with odd number of stars prompts him a safe and lazy option to hit the middle of the scale, which also reduces response quality.

In our case, we kept the original 10-star scale as proposed by the NPC risk-analysis team. Such scale leaves operator no "middle" option, unlike grade "3" on 5-star scale. Indeed, we do not want operators to answer neutrally because, opposite to, e.g., hotel ranking, there is no "neutral" answer other than absence of the edge connecting the two services. Moreover, finer scale makes room for elaborating more precise instructions on self-assessment and answering in the future. As regards the choice of distribution for ξ, it came from papers [5, 6]. The cited authors applied disturbances of moderate scale of one to two stars only.

The main aim of this paper is to evaluate sensitivity of various definitions of service vulnerability, Φ, and of importance aggregation functions, Γ, to errors in user assessment of service impacts.

2. Materials and Methods

2.1. Importance Definitions. There exist a number of recognized and widely known definitions of vertex structural importance that can be used as candidates for Φ. In parlance

of networks, they are usually called *node centralities* [7]. Some of them are trivial ones, like node degree—they are useful but out of scope of this study as they do not consider link weights, i.e., impact values. Some others are related to network flow maximization problems [8]. They also are inappropriate here because software malfunctions, unlike flows, are indivisible, and on the contrary, replicable. This is why we decided to consider the following three ways to calculate service vulnerability:

(i) Φ_{PR} *Page Rank*. Values of $\mathbf{r}$ meet equation $\mathbf{r} = \mathbf{Hr}$, where $\mathbf{H}$ is adjacency matrix $\mathbf{A}$ normalized so that the sum of elements in each column of $\mathbf{H}$ equals one. Vulnerability of a service calculated this way reflects therefore vulnerability of all other services that service depends on. Such was exactly the original idea of web page rank calculation, by Google founders [9]. In our case, a service is a counterpart of a web page. Note, however, that such normalization, necessary from theoretical point of view, weakens impact of vertices with high outdegree. While reasonable for a user clicking through web pages, this assumption does not necessarily hold in case of, e.g., spreading failures, as they may affect dependent services equally strongly, independently of their number.

(ii) Φ_{RC} *Reach Centrality*. Values of $\mathbf{r}$ represent fraction of all services whose operation may affect a given service. To account for service impact, a weighted variant is used [10]. Originally, any v_i affecting v_j increases r_j by $1/(1-|V|)$. In the weighted version, this amount depends on average link weight on the shortest path from v_i to v_j, in relation to average link weight in the graph. With such approach, a kind of weighted impact summation is performed for each service; however, without concern for important structural properties of the graph as, for example, existence of bridges.

(iii) Φ_{MI} *Maximum Input*. Values of $\mathbf{r}$ are solution of the following equation:

$$r_j = \min\left(10, 1 + \frac{1}{10} \max_i a_{ij} r_i\right). \tag{6}$$

The aim of the above formula is to calculate centralities like for page rank; however, taking into account only currently most important factors. Algorithm (6) is repeated until convergence, guaranteed by curbing the outcome within $<1, 10>$ interval, consistent with our rating scheme. Finally, a strongest impact path is created for each dependent service, which identifies most crucial parts of the graph, and service vulnerabilities, accordingly. However, it ignores all relations outside the path, even if they stay close to the path in terms of their importance.

Service vulnerabilities calculated above are based on incoming edges and in fact have the meaning of service susceptibility to failure.

2.2. Aggregation Functions. Vulnerabilities can be aggregated by equation (3) into a single network vulnerability index γ in many ways. Here, we propose three of them:

(i) Γ_{AV}, the mean of $\mathbf{r}$: it represents the total of service vulnerabilities, without regard for their distribution. While providing a good measure of overall vulnerability, it hides the existence of extraordinary vulnerable services in the network.

(ii) Γ_{50}, the median: it represents the typical value of service vulnerability in the network, i.e., it discards extreme values.

(iii) Γ_{MX}, the maximum: contrary to Γ_{50}, the service with biggest vulnerability is picked up, regardless of vulnerability of the other ones.

2.3. Sensitivity of Vulnerability to Self-Assessment Errors. For any instance of reported impact matrix, $\widetilde{A}_m$, we can calculate corresponding $\widetilde{r}_m$ and finally, vulnerability index $\widetilde{\gamma}_m$—using any combination of Φ's and Γ's provided above. Then, we can calculate the difference between vulnerabilities calculated for reported and for real impact values.

$$\delta_m(\Phi, \Gamma) = \widetilde{\gamma}_m - \gamma. \tag{7}$$

In the context of difference between two sets of services, we may introduce yet another measure based on difference in ordering of the most important services there: $\delta_m(\Phi, \Gamma_{\mathrm{L5}})$. It uses Levenshtein distance [11] to compare the contents and order of first five most important services in $\mathbf{r}$ and in $\widetilde{r}$. The Levenshtein distance counts the number of edit operations to apply to one sequence to convert it to another sequence. In our case, five-element sequences are compared. Edit operations are: insertion, deletion, and change of a single element in a sequence. For example, if $\mathbf{r} = [0, 1, 3, 4, 6, 5]$ and $\mathbf{r}_m = [1, 0, 3, 4, 5, 6]$, the five most important services would be $(r_5, r_6, r_4, r_3, r_2)$ and $(r_6, r_5, r_4, r_3, r_1)$, respectively. It takes three operations to transform one set into the other: two for swapping of r_5 with r_6, and one for replacement of r_2 with r_1—and therefore, the edit distance equals three.

2.4. Used Networks. In practice, the service graph G and reported impact values $\widetilde{A}$ are compiled after a laborious process of questioning service operators about their services relationship structure and relationship intensity. A sample real graph of services made this way is presented in Figure 1. Reconstruction of service dependencies between operators is particularly hard, since such information is often considered confidential. Collected data are inherently sensitive because they may serve as well for improving network reliability as for attacking its weakest points. Such observation has been made previously in case of critical infrastructure modeling and holds also for digital services. The papers [12, 13] cover sector-wise interdependency analysis and summarize modeling approaches, respectively. All the authors express their concern about privacy of the collected data; consequently, only a small fraction of interdependencies is

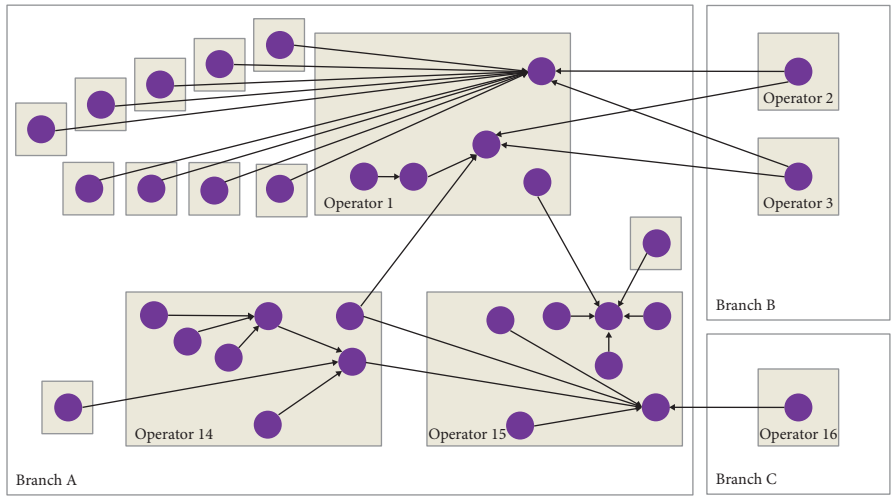

Figure 1: Graph of real dependencies between 33 services run by 17 operators in 3 branches of national economy.

presented in [12]. Similarly, we decided to carry out our study for networks whose operation is partially analogous to the interplay of digital services instead of the real network.

We found that networks of source code dependencies are a close analogy. First, they represent software components, on a much smaller scale though. Second, the dependency between modules can be relatively easily tracked by static code analysis. Third, failure or malfunction of one software module influences the operation of all modules that depend on it, although differently. Fourth, module dependencies in open-source projects appear not in predefined way but represent current needs of programmers, as already reported in [14]. Finally, dependencies between source code modules as well as between essential services can be relatively easily traced, while their intensity can not.

All networks analyzed in this study describe software module dependencies in Javascript (JS) projects available from hosting platform github.com. Dependencies have been found by using the static code analysis tool, Madge http://www.npmjs.com/package/madge. Project properties are given in Table 1. Projects differ in size; moreover, some of them happen to have circular dependencies of the code, which also happens for real digital services. A sample graph of dependencies is shown in Figure 2.

3. Results and Discussion

Formula (7) calculates the vulnerability estimation error for a single realization of $\widetilde{A}$. To assess the error in statistical sense, one would need to calculate analytically how ξ affects $\widetilde{A}$, $\widetilde{r}$, and finally, δ. In this paper, we rather present results of cursory estimation of δ, based on random sampling of δ_m for a number of M samples, $m \in 1, 2, \ldots, M$. We calculate the following statistics from sample distributions of δ:

(i) Mean average absolute error, $\theta_{\mathrm{AE}} = (1/M)\sum_m |\delta_m|$

(ii) Mean average relative error, $\theta_{\mathrm{RE}} = \theta_{\mathrm{AE}}/\gamma$

(iii) Standard deviation of error, $\theta_{\mathrm{AD}} = \mathrm{stdev}(\delta)$

(iv) Standard deviation of error, relative to true value, $\theta_{\mathrm{RD}} = \theta_{\mathrm{AD}}/\gamma$

They all are comprehensive measures of how errors of operators impact estimation affects errors of network vulnerability, given any of the proposed formulas of Φ and Γ.

All the reasoning provided above concerns a single instance of $\mathbf{A}$, whose values are chosen randomly. In order to draw more general conclusions about the properties of chosen combination of Φ and Γ, we need to repeat calculations for a number of test cases. Let us call them experiments—nonzero values of new impact factor matrix $\mathbf{A}$ are chosen and disturbed using equation (4) in each experiment. Finally, all θ's are calculated, accordingly. Sample graphical results from two series of 1,000 experiments each for Airbnb network are given in Figure 3. In all our analyses, from now on, the number of experiments will be equal to the number of samples in each experiment, M.

Figures 3(a) and 3(b) show various characters of vulnerability errors. In some aspects, the two demonstrated examples bear similarity, e.g., γ, and the average of δ is negatively correlated. (Intuitively, the more high-score links in the network, the less important is error by one star in impact estimation by the service operator.) Next, some configurations result in more discrete error distribution—as in case (b) where the switching nature of median manifests in striped dot patterns. Finally, histograms show how much variable are vulnerability errors across experiments. For example, we see that in case (a) they are quite stable, clustered closely around one value, while in case (b) they show much bigger variability.

Results in Figure 3 justify the need for deeper inspection of the nature of observed errors. However, to compare sensitivity of many networks in multidimensional parameter space of Φ's, Γ's, and N's, we have to develop a simpler approach. We propose to calculate and compare average values of, θ's, i.e., $\overline{\theta}_{\mathrm{AE}}$, $\overline{\theta}_{\mathrm{RE}}$, $\overline{\theta}_{\mathrm{AD}}$, and $\overline{\theta}_{\mathrm{RD}}$, over all performed experiments. Such averaged indicators are collected in Tables 2–6, each table for a different project.

Table 1: Properties of projects used for analysis.

Project name	Modules	Number of circular dependencies	Project url
Airbnb	22	0	http://github.com/airbnb/javascript
Fcc	426	18	http://github.com/freeCodeCamp/freeCodeCamp
Nodejs	9507	27	http://github.com/nodejs/node
Omi	475	0	http://github.com/Tencent/omi
React	507	0	http://github.com/facebook/react
Vue	419	8	http://github.com/vuejs/vue

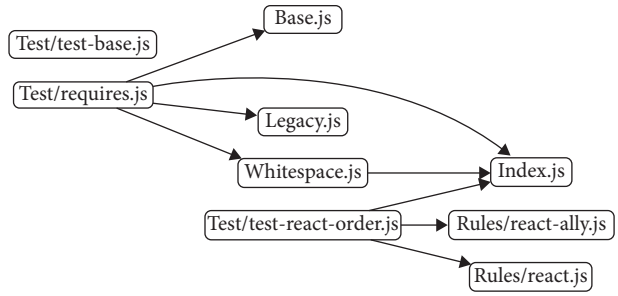

Figure 2: Screenshot of a sample exemplary graph of module dependencies in a part of Airbnb project, displayed by Madge.

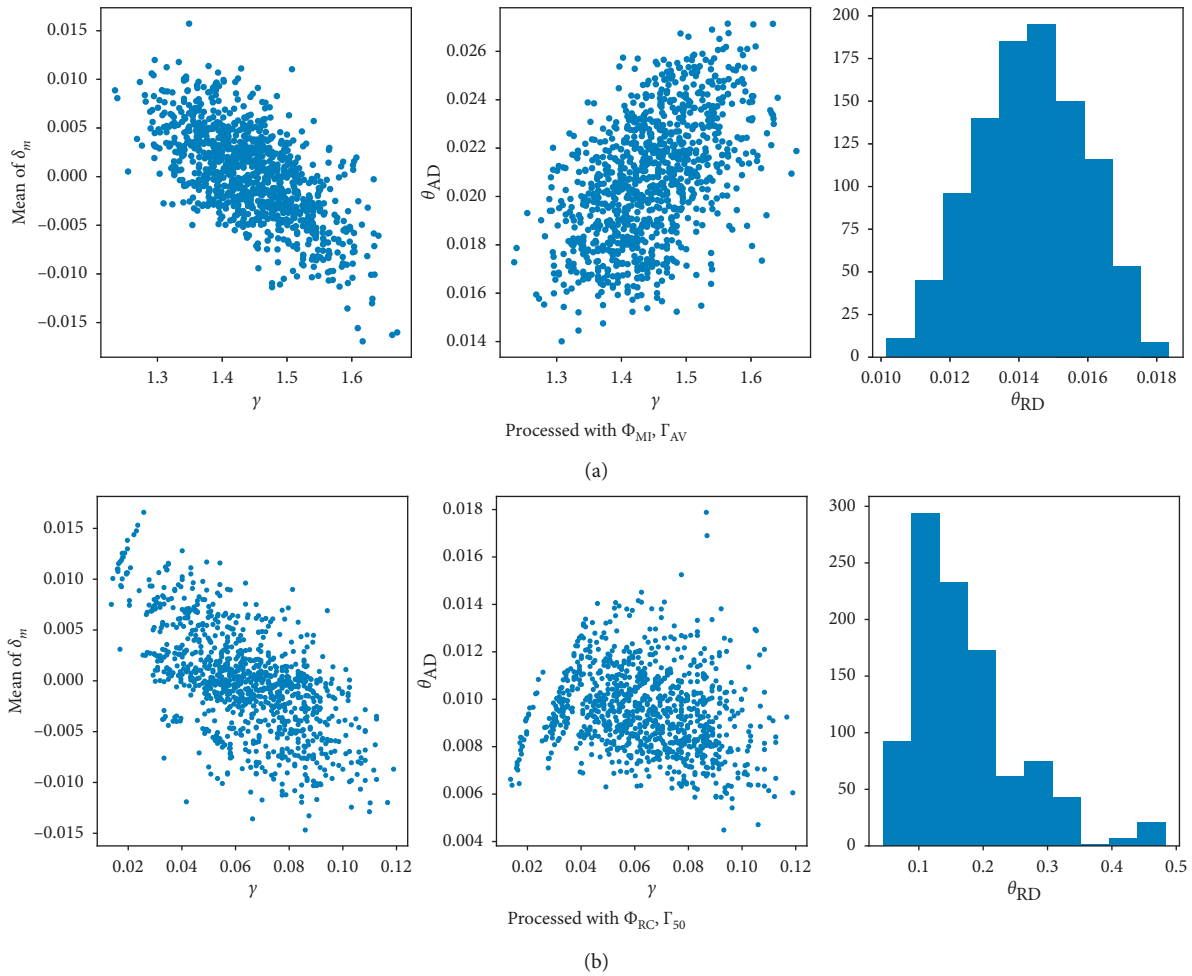

Figure 3: Scatter plots of vulnerability estimation error (left), and standard deviation (middle) vs. true vulnerability. Also (right), standard deviation histogram—for experiments carried out for Airbnb network, with $N = 1$ and different importance and aggregation algorithms (a) and (b).

Table 2: Sensitivity of Airbnb graph, $M = 1000$.

	Φ	Γ_{AV}	Γ_{50}	Γ_{MX}	Γ_{L5}
(a) $N = 1$	Φ_{PR}	$\overline{\theta}_{RE}$: **1.63e-16** $\overline{\theta}_{RD}$: *2.08e-16*	**0.00885** *0.0205*	**0.00162** *0.0102*	$\overline{\theta}_{AE}$: **0.947** $\overline{\theta}_{AD}$: *0.855*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00623** $\overline{\theta}_{RD}$: *0.0125*	**0.0798** *0.172*	**0.0203** *0.0656*	$\overline{\theta}_{AE}$: **1.37** $\overline{\theta}_{AD}$: *1.02*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.00256** $\overline{\theta}_{RD}$: *0.0143*	**0.0184** *0.0456*	**0.0118** *0.0432*	$\overline{\theta}_{AE}$: **1.94** $\overline{\theta}_{AD}$: *1.08*
(b) $N = 2$	Φ_{PR}	$\overline{\theta}_{RE}$: **1.63e-16** $\overline{\theta}_{RD}$: *2.08e-16*	**0.00885** *0.0205*	**0.00162** *0.0102*	$\overline{\theta}_{AE}$: **0.947** $\overline{\theta}_{AD}$: *0.855*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00623** $\overline{\theta}_{RD}$: *0.0125*	**0.0798** *0.172*	**0.0203** *0.0656*	$\overline{\theta}_{AE}$: **1.37** $\overline{\theta}_{AD}$: *1.02*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.00256** $\overline{\theta}_{RD}$: *0.0143*	**0.0184** *0.0456*	**0.0118** *0.0432*	$\overline{\theta}_{AE}$: **1.94** $\overline{\theta}_{AD}$: *1.08*

Table 3: Sensitivity of Fcc graph, $M = 300$.

	Φ	Γ_{AV}	Γ_{50}	Γ_{MX}	Γ_{L5}
(a) $N = 1$	Φ_{PR}	$\overline{\theta}_{RE}$: **2.05e-16** $\overline{\theta}_{RD}$: *2.76e-16*	**0.00541** *0.00936*	**0.00684** *0.028*	$\overline{\theta}_{AE}$: **0.614** $\overline{\theta}_{AD}$: *0.736*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00339** $\overline{\theta}_{RD}$: *0.011*	**0.0178** *0.032*	**0.0114** *0.0286*	$\overline{\theta}_{AE}$: **2.62** $\overline{\theta}_{AD}$: *1.03*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.0108** $\overline{\theta}_{RD}$: *0.0246*	**0.00819** *0.014*	**0.0569** *0.109*	$\overline{\theta}_{AE}$: **3.06** $\overline{\theta}_{AD}$: *1.19*
(b) $N = 2$	Φ_{PR}	$\overline{\theta}_{RE}$: **2.01e-16** $\overline{\theta}_{RD}$: *2.76e-16*	**0.00678** *0.0119*	**0.0119** *0.0451*	$\overline{\theta}_{AE}$: **0.889** $\overline{\theta}_{AD}$: *0.892*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00542** $\overline{\theta}_{RD}$: *0.0172*	**0.0237** *0.0411*	**0.021** *0.0407*	$\overline{\theta}_{AE}$: **3.21** $\overline{\theta}_{AD}$: *0.964*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.0181** $\overline{\theta}_{RD}$: *0.0365*	**0.0113** *0.0186*	**0.11** *0.155*	$\overline{\theta}_{AE}$: **3.74** $\overline{\theta}_{AD}$: *1.08*

Table 4: Sensitivity of Omi graph, $M = 300$.

	Φ	Γ_{AV}	Γ_{50}	Γ_{MX}	Γ_{L5}
(a) $N = 1$	Φ_{PR}	$\overline{\theta}_{RE}$: **2.35e-16** $\overline{\theta}_{RD}$: *3e-16*	**0.00363** *0.00702*	**0.00209** *0.00903*	$\overline{\theta}_{AE}$: **0.596** $\overline{\theta}_{AD}$: *0.694*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.0035** $\overline{\theta}_{RD}$: *0.00893*	**0.0171** *0.0285*	**0.00533** *0.0179*	$\overline{\theta}_{AE}$: **1.55** $\overline{\theta}_{AD}$: *0.936*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.00143** $\overline{\theta}_{RD}$: *0.00698*	**0.00711** *0.0118*	**0.025** *0.0619*	$\overline{\theta}_{AE}$: **3.31** $\overline{\theta}_{AD}$: *1.01*
(b) $N = 2$	Φ_{PR}	$\overline{\theta}_{RE}$: **2.31-16** $\overline{\theta}_{RD}$: *3e-16*	**0.00501** *0.00909*	**0.00391** *0.0146*	$\overline{\theta}_{AE}$: **0.901** $\overline{\theta}_{AD}$: *0.833*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00566** $\overline{\theta}_{RD}$: *0.0138*	**0.022** *0.0366*	**0.00845** *0.0283*	$\overline{\theta}_{AE}$: **1.97** $\overline{\theta}_{AD}$: *0.928*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.0033** $\overline{\theta}_{RD}$: *0.0113*	**0.00978** *0.0166*	**0.0472** *0.0849*	$\overline{\theta}_{AE}$: **3.91** $\overline{\theta}_{AD}$: *0.906*

Table 5: Sensitivity of React graph, $M = 300$.

	Φ	Γ_{AV}	Γ_{50}	Γ_{MX}	Γ_{L5}
(a) $N = 1$	Φ_{PR}	$\overline{\theta}_{RE}$: **2.1e-16** $\overline{\theta}_{RD}$: *2.43e-16*	**0.00473** *0.00867*	**0.00595** *0.0264*	$\overline{\theta}_{AE}$: **0.418** $\overline{\theta}_{AD}$: *0.524*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00307** $\overline{\theta}_{RD}$: *0.0104*	**0.0233** *0.0405*	**0.0145** *0.0321*	$\overline{\theta}_{AE}$: **2.52** $\overline{\theta}_{AD}$: *1.03*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.0137** $\overline{\theta}_{RD}$: *0.0419*	**0.0107** *0.0183*	**0.0319** *0.0549*	$\overline{\theta}_{AE}$: **3.11** $\overline{\theta}_{AD}$: *1.11*
(b) $N = 2$	Φ_{PR}	$\overline{\theta}_{RE}$: **2.12e-16** $\overline{\theta}_{RD}$: *2.43e-16*	**0.00611** *0.0114*	**0.00921** *0.0444*	$\overline{\theta}_{AE}$: **0.561** $\overline{\theta}_{AD}$: *0.657*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00495** $\overline{\theta}_{RD}$: *0.0163*	**0.0285** *0.0541*	**0.0213** *0.0452*	$\overline{\theta}_{AE}$: **3.08** $\overline{\theta}_{AD}$: *0.996*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.0268** $\overline{\theta}_{RD}$: *0.0623*	**0.0128** *0.0245*	**0.0455** *0.074*	$\overline{\theta}_{AE}$: **3.63** $\overline{\theta}_{AD}$: *1.03*

Table 6: Sensitivity of Vue graph, $M = 500$.

	Φ	Γ_{AV}	Γ_{50}	Γ_{MX}	Γ_{L5}
(a) $N = 1$	Φ_{PR}	$\overline{\theta}_{RE}$: **2.05e-16** $\overline{\theta}_{RD}$: *2.71e-16*	**0.00524** *0.0103*	**0.00452** *0.0205*	$\overline{\theta}_{AE}$: **1** $\overline{\theta}_{AD}$: *0.884*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.0046** $\overline{\theta}_{RD}$: *0.0136*	**0.016** *0.0302*	**0.0156** *0.0319*	$\overline{\theta}_{AE}$: **2.84** $\overline{\theta}_{AD}$: *1*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.0101** $\overline{\theta}_{RD}$: *0.0312*	**0.00949** *0.0168*	**0.0053** *0.12*	$\overline{\theta}_{AE}$: **3.07** $\overline{\theta}_{AD}$: *1.11*
(b) $N = 2$	Φ_{PR}	$\overline{\theta}_{RE}$: **1.98e-16** $\overline{\theta}_{RD}$: *2.72e-16*	**0.00708** *0.0135*	**0.00759** *0.033*	$\overline{\theta}_{AE}$: **1.38** $\overline{\theta}_{AD}$: *0.944*
	Φ_{RC}	$\overline{\theta}_{RE}$: **0.00711** $\overline{\theta}_{RD}$: *0.0211*	**0.0206** *0.0404*	**0.0247** *0.0431*	$\overline{\theta}_{AE}$: **3.44** $\overline{\theta}_{AD}$: *0.944*
	Φ_{MI}	$\overline{\theta}_{RE}$: **0.0213** $\overline{\theta}_{RD}$: *0.0506*	**0.0112** *0.0228*	**0.104** *0.181*	$\overline{\theta}_{AE}$: **3.74** $\overline{\theta}_{AD}$: *1.01*

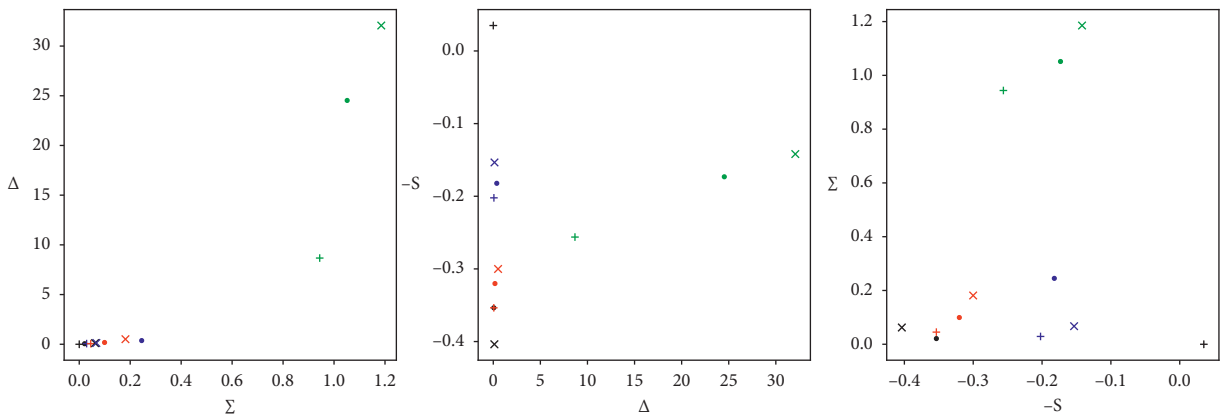

Figure 4: Values of Δ, the negative of S, and Σ presented in pairs in separate graphs. Shapes denote methods used for importance calculation: Φ_{PR}—plus; Φ_{RC}—dot; Φ_{MI}—cross. Colors denote aggregation methods used: Γ_{AV}—black; Γ_{50}—blue; Γ_{MX}—red; Γ_{L5}—green.

The figures given in Tables 2–6 cover all combinations of five graphs, three importance indices Φ, four importance aggregation functions Γ, and two amplitudes of estimation error N. Basically, we search this space to find valuable combinations of Φ's and Γ's. A valuable combination is characterized by

(i) Small total error Δ for all considered projects and values of N—we want the approach to be independent of graph structure

(ii) Big sensitivity S to change of N, for all projects (pick the worst case)—we want operators' errors of estimation to really influence the value of overall metrics θ

(iii) Small standard deviation Σ of error, for all projects (pick the worst case)—we want small variance of θ's, in general

Candidate combinations of Φ and Γ should therefore be in general tolerant to imprecise information provided by operators, but at the same time, sensitive to the scale of such lack of precision. Moreover, it is desirable that errors in network vulnerability calculated by such combination do not vary widely. We check the last two requirements with respect to the worst results found for the analyzed projects. Results of such three-criteria scoring are presented in Figure 4, projected on three planes. The axes have been selected or adjusted so that markers located near an axis correspond to combinations that perform better. Visual comparison provided in Figure 4 does not determine strictly the optimum combination, but makes it possible to observe that, in general, performance indices do not vary widely—at least so that using linear axis scaling will do to reveal differences. Secondly, markers get clustered mainly with respect to their color, which means that the choice of aggregation method Γ is more important than the choice of algorithm for importance index calculation.

As analyzed combinations form a cloud in 3D space, we may find a Pareto front, i.e., a set of nondominated combinations. They are

(i) (Φ_{RC}, Γ_{AV})—the average of reach centrality

(ii) (Φ_{PR}, Γ_{AV})—the average page rank

(iii) (Φ_{PR}, Γ_{AV})—the median of page rank

(iv) (Φ_{PR}, Γ_{MX})—the maximum of page rank

(v) (Φ_{MI}, Γ_{AV})—the average of maximum input importance

4. Conclusions

It should be reminded that research reported here is done in context of a large project aiming to build a nation-wide

model of critical services network. While integrity of the resulting graph can be obtained by careful automated inspection of questionnaires filed by service operators, the estimated reported impact between services will be biased and inherently erroneous. Therefore, it was worth to study sensitivity of some candidate synthetic metrics of overall network vulnerability with respect to incorrect input. We felt it correct to use networks of software module dependencies because of their functional and structural similarity to network of critical services, let alone that such real networks will probably remain confidential.

The study shows that all three proposed formulas for individual service vulnerability calculation are valuable. This is rather a positive observation, as each of them has its own specifics and can be used under various circumstances. Also, almost all proposed ways of vulnerability aggregation into a single vulnerability index are useful (except the Levenshtein distance, which shows much variation and has turned out to be useless). Naturally, combinations of formulas appropriate for capturing "extreme" phenomena, as (Φ_{MI}, Γ_{MX}), will have show variability.

The main takeaway is that it is safe to apply mean or median aggregation of individual service vulnerability, whatever is the formula for importance calculation. Such aggregated value may serve as a single, comprehensive vulnerability index. Note that being robust to errors in graph edge weights, it will be affected by major structural graph changes—e.g., edge removal as result of real-time detected failure. Our previous work has shown that networks of autonomous systems (AS) can be really badly affected by just one link failure, contrary to widespread belief in Internet robustness [15].

One should remember that results reported here were based on the sound assumption of analogy between critical services and software modules. This assumption will eventually get verified in practice, once the national cybersecurity platform is operational and filled with data. We look forward to compare properties of vulnerability calculation formulas calculated here by random sampling with careful expert judgment and postmortem analyses for real services graph.

Data Availability

The open source code used to support the findings of this study is publicly available on http://github.com and can be downloaded and processed with tools indicated in this paper. The proprietary Python code created by the author to analyze data used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work presented in this paper has been supported by the Polish National Centre for Research and Development grant (CYBERSECIDENT/369195/I/NCBR/2017).

References

[1] The European Commission, *The Directive on Security of Network and Information Systems*, The European Commission, Brussels, Belgium, 2016.

[2] J. Hingant, M. Zambrano, F. J. Pérez, I. Pérez, and M. Esteve, "Hybint: a hybrid intelligence system for critical infrastructures protection," *Security and Communication Networks*, vol. 2018, Article ID 5625860, 13 pages, 2018.

[3] G. Settanni, F. Skopik, Y. Shovgenya et al., "A collaborative cyber incident management system for european interconnected critical infrastructures," *Journal of Information Security and Applications*, vol. 34, pp. 166–182, 2017.

[4] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, *Computer Security: Principles and Practice*, Pearson Education, Upper Saddle River, NJ, USA, 2012.

[5] M. Medo and J. R. Wakeling, "The effect of discrete vs. continuous-valued ratings on reputation and ranking systems," *EPL (Europhysics Letters)*, vol. 91, no. 4, Article ID 48004, 2010.

[6] W. W. Moe and M. Trusov, "The value of social dynamics in online product ratings forums," *Journal of Marketing Research*, vol. 48, no. 3, pp. 444–456, 2011.

[7] Networkx Manual, Centrality Methods Reference, 2019, https://networkx.github.io/documentation/stable/reference/algorithms/centrality.html.

[8] U. Brandes and D. Fleischer, "Centrality measures based on current flow," in *Annual Symposium on Theoretical Aspects of Computer Science*, pp. 533–544, Springer, Berlin, Germany, 2005.

[9] L. Page, S. Brin, R. Motwani, and W. Terry, "The pagerank citation ranking: bringing order to the web," Tech. Rep., Stanford InfoLab, Stanford, CA, USA, 1999.

[10] E. Mones, L. Vicsek, and T. Vicsek, "Hierarchy measure for complex networks," *PLoS One*, vol. 7, no. 3, Article ID e33799, 2012.

[11] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[12] C.-N. Huang, J. J. H. Liou, and Y.-C. Chuang, "A method for exploring the interdependencies and importance of critical infrastructures," *Knowledge-Based Systems*, vol. 55, pp. 66–74, 2014.

[13] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability Engineering & System Safety*, vol. 121, pp. 43–60, 2014.

[14] M. Kamola, "How to verify conway's law for open source projects," *IEEE Access*, vol. 7, pp. 38469–38480, 2019.

[15] K. Mariusz and A. Piotr, "Network resilience analysis: review of concepts and a country-level. case study," *Computer Science*, vol. 15, no. 3, p. 311, 2014.