Date of publication DEC-31, 2021, date of current version SEP-02, 2021.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.20.4.2446

The Measurement of Popularity and Prevalence of Software Vulnerability

YULIIA TATARINOVA[1,2], OLHA SINELNIKOVA[2]

[1]Kharkiv National University of Radio Electronics, Nauky Ave. 14, 61103, Kharkiv, Ukraine (e-mail: yullia.tatarinova@gmail.com)
[2]Samsung Electronics Ukraine Company LLC, L. Tolstogo St., 57, 01033 Kyiv, Ukraine (e-mail: ol.sinelnikova@gmail.com)

Corresponding author: Yuliia Tatarinova (e-mail: yullia.tatarinova@gmail.com).

ABSTRACT Prioritizing bug fixes becomes a daunting task due to the increasing number of vulnerability disclosure programs. When making a decision, not only the Common Vulnerability Scoring System (CVSS) but also the probability of exploitation, the trend of particular security issues should be taken into account. This paper aims to discuss the sources and approaches for measuring degree of interest in a specific vulnerability at a particular point in real-time. This research presents a new metric and estimation model which is based on vulnerability assessment. We compared several techniques to determine the most suitable approach and relevant sources for improving vulnerability management and prioritization problems. We chose the Google Trend analytics tool to gather trend data, distinguish main features and build data set. The result of this study is the regression equation which helps efficiently prioritize vulnerabilities considering the public interest in the particular security issue. The proposed method provides the popularity estimation of Common Vulnerabilities and Exposures (CVE) using public resources.

KEYWORDS trend analysis; CVE; vulnerability assessment; impact evaluation.

I. INTRODUCTION

INFORMATION security (IS) news feeds are increasingly being updated with information on new vulnerabilities in various products. Many software development companies, huge corporations or startups in the field of information technology use a large number of third-party software products. At the same time, it is not always possible to use the latest updated versions of the software. This is due to strict dependencies on a specific version and its functionality, lack of documentation, increased overhead costs for testing and updating, lack of human resources and high risks for business in case of an error. Moreover, in most cases, it is impossible to update all existing and known vulnerabilities based on the above difficulties. Many companies apply patches only to the most critical vulnerabilities. For example, having n vulnerabilities, a company can only allocate resources and time to fix m vulnerabilities, where $m \ll n$. Since fixing all n vulnerabilities is not economically viable in terms of business and profit.

Hence the following requirements arise: accurate vulnerability assessment regarding the system used, issue severity rating, prioritizing patches, etc. Recently, information from open sources is also included in a set of factors for making final decisions. The manifestations of public interest in information security in the most risk-critical vulnerabilities, as well as mentioning in the media, forums, and private chats, are also gaining great popularity. All this bears reputational losses for the software manufacturer. At the same moment, there is no technology or tool, that allows you to accurately and reliably determine the degree of community interest. One of the main reasons is that this area of activity is specific and niche not only in the whole world but also in the field of information technology.

Let us state the problem before moving on to a previous work review. Our objective is to design an impact weaknesses evaluation system that can automatically estimate disclosed security flaw impact (W) on the end-point product (P) [1]. The main system architecture and feature set description and extraction process were presented in [2]. A

similar technology with [1] was introduced in [3]. An important part of our study is that the relevance of specific vulnerability decreases over time, giving way to newer ones. This issue was not considered in [3] and in other risk assessment systems. Thus, the Common Vulnerability Scoring System (CVSS) [4] is not fully applicable. One of the main weaknesses of the CVSS is that score computed once and the value is used for a long period of time. Therefore, it is necessary to determine the degree of relevance of the vulnerability at a particular point in time ($Trend(CVE_i)$). The relevance of the vulnerability can be expressed through the degree of interest of the security community in it.

In this paper, for the first time, ways of searching and identifying possible sources of the degree of community interest in vulnerability are presented. A method for their assessment is also proposed for implementation in an integrated system of automatic vulnerability assessment, which was presented in [1].

II. RELATED WORK

Existing materials on the topic of trends and popularity vulnerability can conditionally be divided into the following categories:

1. Lists of the most critical vulnerabilities for a given period of time.
2. An overview of trends in vulnerability types, weaknesses, exploits and software.

Let us look at each item separately and in detail.

A. LISTS OF THE MOST CRITICAL VULNERABILITIES

Sources for this category are blogs, web articles, news feeds, and popular information security sites. The information is often presented in the form of a small list. Presented list has next format: issue identifier (Id) and a short description ($Descr$) [5, 6].

The most popular are the following sources:

- OWASP top 10 [7] 2017. This source contains a list of the most critical web application security risks. The list is rarely updated by the community (approximately once every 3– 4 years). It may be noted that the bulk of the list has remained unchanged for many years. This source also provides a similar list of typical Internet of Things vulnerabilities and mobile part [8].
- The most exploited vulnerabilities for 2018 were described in [9] and [10]. Other blogs, articles, or news posts are more likely to repeat the same list presented. We analyzed the selections of posts and messages and concluded that the main criteria for rating vulnerabilities are Common Vulnerability Scoring System (CVSS), popularity and prevalence of affected products (P_{list}), ability and methods of exploitation. The resulting rating may contain two types of information: types of the most common and exploited vulnerabilities and threats (e.g., XSS, weaknesses in authentication or access control) and a list of specific vulnerabilities with identifiers (CVE-2016-0189, CVE-2017-0199).

B. SOURCES DESCRIBING VULNERABILITY TRENDS

Identification of trends in vulnerability parameters allows you to visually see the overall picture of risks, determine the most likely attacks and a protection strategy against hacking. To this end, many companies in the field of information security produce annual reports with statistics and forecasts. The vulnerability information is collected and analyzed manually by experts in most cases.

In [5], the team scanned data from dozens of channels and security sources, and also explored sites in the dark network; checked and improved data using automatic as well as manual analysis. At the same time, analysts added their knowledge about attack trends, cyber events. The resulting report is divided into sections such as the most exploited products, popular attacks, threats, etc.

In [11] and [12], the authors provided general statistics and the distribution of vulnerabilities by type at the source code stage, CVSS level, and source of vulnerabilities. The time frames in the article refer to 2008 – 2016.

Paper [13] contains a fairly good overview of the statistics and trends of the most popular "Bug Bounty" programs. The authors described not only the operating schemes of reward programs for vulnerabilities found, but also provided well-chosen and structured data on the types of vulnerabilities, the time of repair, and the products.

III. CVE TREND INVESTIGATION APPROACH

Existing articles on vulnerability trends mainly describe statistics on the frequency and type of vulnerabilities, types of attacks, source and other parameters. However, this does not take into account the frequency of the request for a particular vulnerability. For the first time, the idea of introducing a time characteristic for a specific vulnerability was presented in [1]. When a new widely exploited vulnerability is published, there is an increased interest in it from attackers and security experts. Over time, for some vulnerabilities discovered, demand falls, for others it remains stable for a long time. It is the surge of community interest and the degree of its severity at a particular point in time that is never taken into account in many widely used models and methods for assessing vulnerability risk.

In this subsection, we describe our previous studies and try to find data sources to obtain free information on vulnerability trends. The first attempts to find various methods for detecting and evaluating this characteristic were shown in [2]. We defined the temporary variable $Rel\ (t)$ ($Trend$). We have to find sources, which are publicly available and easily discoverable. This characteristic is very significant for the study since it determines the degree of community interest in information security relative to time for a particular vulnerability. Each vulnerability (CVE_i) in the NIST database (DB) [14] contains two timestamps: date of creation in the database ($D_{created}$) and the date of the last change ($D_{modified}$). It should be noted that these variables are not always present, and some information about CVE_i may be distributed before updating the database. Having only these two parameters available, it is impossible to

hypothesize and evaluate the degree of interest ($Trend$).

The study of blogs, chats, and forums on information security becomes a very promising direction. However, such research requires the most in-depth knowledge in the field of processing of texts written in natural languages and significant human resources in terms of marking up unstructured texts. Moreover, such an in-depth analysis of this parameter is redundant in the context of its application in the system presented in [1].

When investigating this issue, the following sources of information were identified for constructing and analyzing the above-described Trend characteristics:

1. Internet Archive Search [15]. It collects website traffic statistics. The archive provides open and free access to its databases. The content of web pages is occasionally recorded using a bot. The system accepts a link as an input and displays a page cache map. At the time in the study in [2], this system did not show how many times the page was updated. It was also impossible to quantify the indicators of changes. Based on this, this source of information was rejected in the construction of the hypothesis. Also, a significant drawback was the small number of cached pages in general for the vulnerability database. At the same time, at the time of writing, the service has additional functionality that allows you to identify content changes step by step.
2. Google trends [16]. Google's open web application that shows how often a particular term is searched concerning the total volume of search queries in different regions of the world and different languages. Also, Google Trends displays news related to search phrases, superimposing them on a graph showing how new events affect search popularity. Previously, this service was already successfully used in the healthcare sector to track the incidence of influenza [17], thus proving the service's ability to analyze the popularity and seasonality of search queries in specific areas. To obtain data in the context of the current task, a vulnerability identifier (Id) and a publication date ($D_{created}$) are sent to the service input and a time series with integer search intensities $T(CVE_i)$ is output. Values range from 0 to 100. The numbers indicate the level of interest in the topic relative to the highest value in the time series for a specific region and time period. 100 points mean the highest level of request popularity, 50 - request popularity level, half as low as in the first case. 0 points means a location for which there is insufficient data on the request in question [1]. The step length is 1 day for 2019 vulnerabilities and 1 week for the rest.
3. Vulmon [18]. A vulnerability search system that contains statistics on the number of hits within its limits. Its results can only be used to clarify the information received. Often the result of its use is incomplete or data is not available.

IV. CVE TREND EVALUATION PROCESS

This section consists of two parts: determining the characteristics and parameters of a trend and an automated method for estimating the CVE of a trend.

A. MAIN CHARACTERISTICS EXTRACTION

To analyze the resulting trend, it is necessary first of all to formalize it, namely, to determine the parameters specific to each case. For the analysis and determination of trend characteristics, the methodology that was proposed in [19] was used as the basis. As an example, Fig. 1 shows the dynamics of popularity and some characteristics of the trend CVE-2017-12542. A complete list of characteristics, as well as extraction methods, is presented in Table 1.

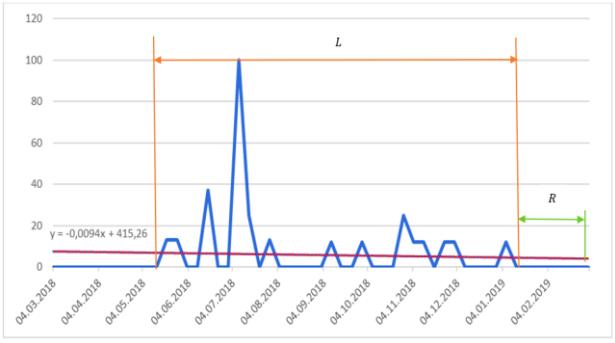

Figure 1. Dynamics and main features of trend CVE-2017-12542

Table 1. Characteristics for assessing the trend of vulnerability

Parameter Name	Value Range	Extraction Method
Dynamics, ***D***	[-90; 90]	Eq. (1)
Duration, L	[0; 100]	Eq. (2)
Mutability, σ	[0; 100]	Standard deviation
Remoteness, ***R***	[0; 100]	Eq. (3)
Frequency, F	[0; n]	Sum of non-zero values

We took into account the following factors:

1. Dynamics or rate of change. Dynamics can be described by a straight line, the slope of which characterizes the tendency to increase or decrease interest. The quantitative value may be within the interval of the tilt angle [-90; 90]. The value is obtained based on the coefficients of the linear regression equation, namely the angular coefficient (b1):

$$D = 57{,}2958 * \arctan(b1). \quad (1)$$

2. Duration. Measured by the time interval from the first to the last burst of (non-zero) values. The result is a percentage of the duration to the total length of the time series.

$$L = 100 * \left(\frac{peak_{last} - peak_{first}}{n}\right), \quad (2)$$

where n – time series length, $peak_{last}$ – index of the last

non-zero value, $peak_{first}$ – index of the first non-zero value.

3. Mutability. Reflects the distribution of the obtained values in the time series. In this paper, it is proposed to use the standard deviation to calculate the variance, as the most common indicator of the dispersion of random values relative to its mathematical expectation.
4. Remoteness. It characterizes in percentage terms how far the last surge is distant from the current date:

$$R = 100 * \left(\frac{peak_{last}}{n}\right). \quad (3)$$

5. Frequency. The total number of non-zero values in the trend. Most of the obtained trends contain a fairly small number of bursts.

B. CVE TREND ESTIMATION METHOD

The aim of this work is to obtain a method that is able to automatically assess the significance of the level of popularity of a vulnerability relative to a set and present the result in quantitative terms. The obtained result can be used for further calculations in [1].

For a given set of trend characteristics ($Trend_{vars}$) (see Table 1), it is required to construct a dependence from the input set of values to the output ($Trend_{score}(CVE_i)$).

The ranges of values that can take input variables are given in Table 1. For each CVE_i we collect from [15] data with $Trend$. As a result, we obtained more than 3,000 non-empty vulnerability trends, which are represented as time series of data points. Manually estimation of each $Trend(CVE_i)$ it is a time consuming and error-prone problem. So, it was decided to automate this task with a modeling technique. To automatically estimate the trend value based on the calculated characteristics and establish the relationship between the characteristics given in Table 1 and the output trend ($Trend_{score}$), a polynomial linear regression model is used. We employ a logistic regression because it is more intuitive and easily implemented.

We are looking for the dependence on trend variables in the form of regression. All extracted variables we include into a logistic regression model:

$$Trend(CVE_i) = f(\alpha_0 + \alpha_d D_i + \alpha_l L_i + \alpha_\sigma \sigma_i + \alpha_r R_i + \alpha_f F_i + \varepsilon_i) \quad (4)$$

V. RESULTS AND FEATURE WORK

For the experiment, we need to collect data set for the period 2016 – 2018. This data set was divided into 3 according to each year. For each CVE_i we send request to Google Trend service [16] with CVE id (CVE_{id}), date of creation ($D_{created}$) and current date ($D_{current}$). Response contains trend ($Trend_i$), but in most cases it is empty. Then, we calculate a vector of characteristics ($Trend_{vars}$) for each non-empty obtained trend. Table 2 represents data sets overview.

Table 2. Data sets overview and coefficient of determination

Year	2017	2018	2019	Total
Number of CVEs	14714	16556	12174	43444
Training data set	184	118	98	403
Full dataset size	941	824	293	2058
R^2	0.398	0.737	0.5779	0.489

Next step was to produce training sets for the declared model. We need to select data points for the training set, which are uniformly distributed in the resulting set.

Assume, that the vector of characteristics ($Trend_{vars}$) is a five-dimensional point (T_i) for each trend ($Trend_i$). To select points for dataset, we need to represent high-dimensional dataset in a low-dimensional space of two dimension. Obtained result we can visualize and select point from each region. We used t-SNE algorithm for dimensionality reduction [20]. For each data set by year, we run t-SNE algorithm and marked data point according to the expert evaluation metrics until all visualized regions were covered. Fig. 2 shows the intermediate stage of the process and data visualization markup. Evaluated trends are represented as blue dots, untagged dots are marked up by blue color.

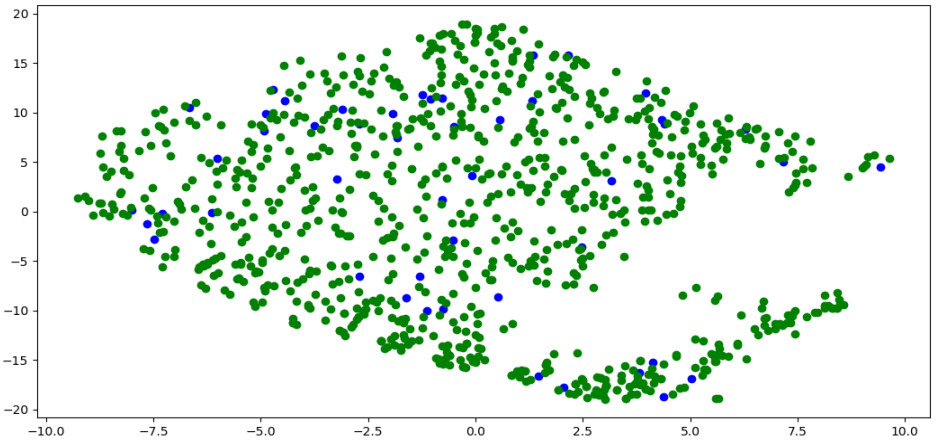

Figure 2. Intermediate stage of the data markup process using t-SNE

For each testing data set by year, logistic regression analysis was used to model output estimating trend score (eq. (4)). Next step, we evaluate obtained models by each year with different datasets and mixed dataset. According to Table 2, it is clear that the best results we obtained while using model, obtained from 2018 year dataset. Table 3 shows that the best results were obtained with the use of 2018 year dataset model. Fig. 3 displays graphical summary of applying best model to the testing dataset. Final logistic regression coefficients could be described as:

$$Trend(CVE_i) = -0.96 * D_i + 1.32 * L_i - 2.37 * \sigma_i - 1.41 * R_i + 14.13 * F_i. \quad (5)$$

Table 3. Data sets overview and coefficient of determination

Training / Testing	2017	2018	2019
2017	12.9	7.94	10.216
2018	12.425	7.38	9.953
2019	12.86	8.092	10.153

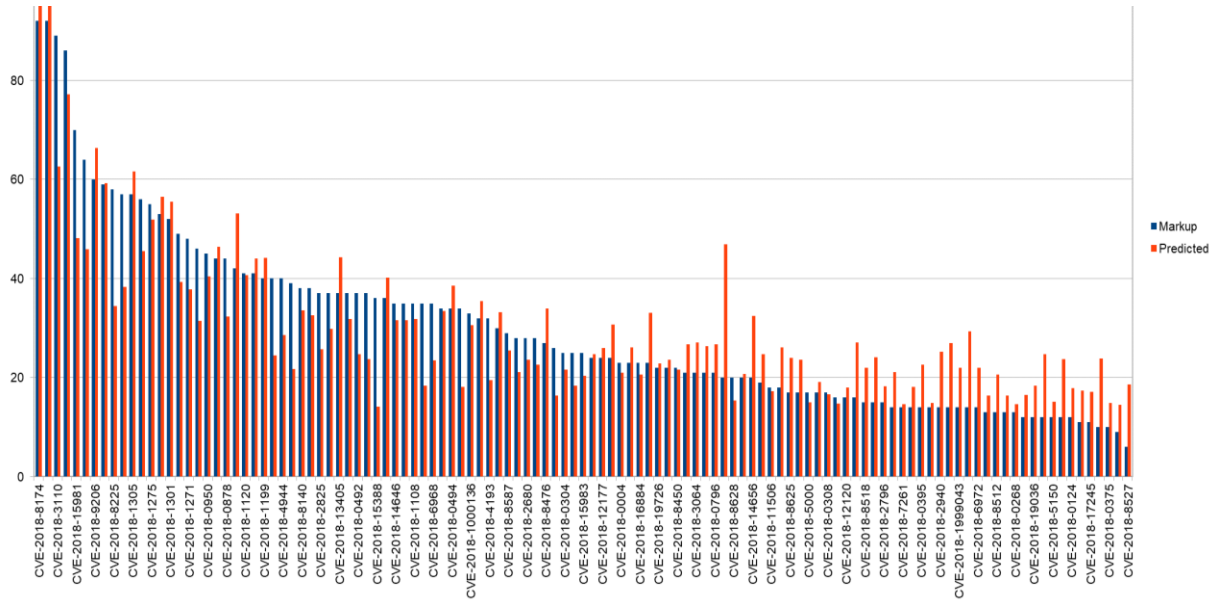

Figure 3. Histogram of training and verification of the obtained model

Thus, we have built a system for automatic reevaluation of vulnerabilities from the CVE database. This method is intended to simplify the SDL procedure during development, to rank vulnerabilities in ascending order of risks, determine the most critical and state-of-the-art issues.

There are several possible ways, which could improve our results. For example, you can improve training datasets and retrain model again. This occurs due to improper data markup. Apply Internet Archive Search approach and extend trend characteristics set. Next step is to embed the regression coefficients into the evaluation system [1] to add dependence of the final result relevance on time.

References

[1] Yu. Tatarinova, "AVIA: Automatic vulnerability impact assessment on the target system," *Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, 2018, pp. 364-368. https://doi.org/10.1109/DSMP.2018.8478519.

[2] Yu. Tatarinova, and O. Sinelnikova, "Extended vulnerability feature extraction based on public resources," *Theoretical and Applied Cybersecurity*, vol. 1, no. 1, pp. 59-67, 2019. https://doi.org/10.20535/tacs.2664-29132019.1.169085.

[3] J. Jacobs, S. Romanosky, B. Edwards, M. Roytman, & I. Adjerid, "Exploit Prediction Scoring System (EPSS)," 2019. arXiv preprint arXiv:1908.04856.

[4] FIRST project, Common Vulnerability Scoring System SIG, [Online]. Available at: https://www.first.org/cvss/

[5] Skybox Research Lab: vulnerability report, [Online]. Available at: https://lp.skyboxsecurity.com/rs/440-MPQ-510/images/Skybox_Report_Vulnerability_and_Threat_Trends_2019.pdf

[6] Security trails: Top CVEs exploited in the wild, [Online]. Available at: https://securitytrails.com/blog/top-cves-exploited-in-the-wild

[7] OWASP Top Ten Project, [Online]. Available at: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf

[8] OWASP Internet of Things Project, [Online]. Available at: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project

[9] SecurityTrails, blog, [Online]. Available at: https://securitytrails.com/blog/top-cves-exploited-in-the-wild

[10] Securityweek, [Online]. Available at: https://www.securityweek.com/top-vulnerabilities-exploited-cybercriminals

[11] D. R. Kuhn, M. S. Raunak, & R. Kacker, "An analysis of vulnerability trends, 2008-2016," *Proceedings of the 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, July 2017, pp. 587-588. https://doi.org/10.1109/QRS-C.2017.106.

[12] R. Kuhn, M. Raunak, and R. Kacker, "It doesn't have to be like this: Cybersecurity vulnerability trends," *Professional*, vol. 19, issue 6, pp. 66-70, 2017. https://doi.org/10.1109/MITP.2017.4241462.

[13] J. Ruohonen, and L. Allodi, "A bug bounty perspective on the disclosure of web vulnerabilities," 2018. arXiv preprint arXiv:1805.09850.

[14] National Vulnerability Database, [Online]. Available at: https://nvd.nist.gov/
[15] Wayback Machine, [Online]. Available at: https://archive.org/
[16] Google trends, [Online]. Available at: https://trends.google.com/trends
[17] J. Ginsberg, et al., "Detecting influenza epidemics using search engine query data," Nature, vol. 457, no. 7232, pp. 1012-1014, 2009.
[18] Vulmon, [Online]. Available at: https://vulmon.com/
[19] J. Kacprzyk, A. Wilbik, S. Zadrozny, "Linguistic summarization of trends: A fuzzy logic based approach," *Proceedings of the 11th International Conference Information Processing and Management of Uncertainty in Knowledge-based Systems*, July 2006, pp. 2166-2172.
[20] L. van der Maaten, and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579-2605, 2008.
[21] J. Ruohonen, S. Hyrynsalmi, and V. Leppanen, "Modeling the delivery of security advisories and CVEs," *Computer Science and Information Systems*, vol. 14, issue 2, pp. 537–555, 2017. https://doi.org/10.2298/CSIS161010010R.

YULIIA E. TATARINOVA, Master in Cybersecurity Technology from Kharkiv National University of Radio Electronics, Lead Engineer at Samsung R&D Institute Ukraine. Areas of scientific interests: security automation, software assessment, artificial intelligence.

OLHA I. SINELNIKOVA, Doctor of Philosophy, Senior Engineer at Samsung R&D Institute Ukraine. Olha is involved in development solutions for security protection of device and server side components: intrusion and abnormal detection system, self-healing systems.