Towards Cross Project Vulnerability Detection in Open Source Mobile Applications

SCT417-C004-1275/2017: - PETER B. OBIRIA

PhD in Information Technology

SCI 4208: Scientific Writing and Research Ethics- **TASK 1**

Abstract

Open source software avails its design publicly posing enormous risk to enterprises. Early vulnerabilities detection is an important step towards building secure open source mobile applications. Cross project vulnerability detection plays significant roles in appraising the most likely vulnerable components. However, current vulnerability detection models that identify vulnerable components are focused on reducing time and effort needed to secure software using process or product metrics and machine learning techniques. Conversely, little effort has been spent to deliver modalities to choose the training data for cross project vulnerability detection. Therefore, we present an empirical study and experiment to demystify cross project open source mobile app vulnerability detection using clustering, a data mining technique to group similar data. The study will be conducted on publicly available dataset of several open source mobile applications and in the context of cross project vulnerability detection.

1. Introduction

Open source software avails its source code for anyone to inspect, modify, and / or enhance. Its design is publicly accessible posing enormous risk to enterprise security despite their many benefits. Research has shown that criminals can use brute-force password attacks against the popular open source Magento ecommerce platform, leveraging a compromised access to scrape credit card records and install malware focused on crypto currency mining (Korolov, 2018). Conversely, software development is a complex endeavor that contends with limited resources potentially causing software to behave in an unforeseen manner (Fenton & Bieman, 2014). Today, mobile devices have become ubiquitous with a recent survey of enterprise and small business workers indicating that just 3% of organizations ban their employees from using personal iPads or iPhones for business use, and only 7% ban Android devices (Harvey, 2015). Further, the study

show that 40% of organizations provide iPhones for more than a quarter of employees, and 25% provide Android-based smartphones. However, modern users are concerned about the security of the applications (Apps) they use. Free mobile apps pose a serious threat to privacy because of their ability to capture large amounts of user information, a study has revealed. According to findings from Juniper Networks, free mobile applications are 401% more likely to track user location and 314% more likely to access user address books than paid-for apps (Warwick, 2012). Majority of the analyzed apps had access to the internet, which could provide a revenues to expose data to be transmitted from the device.

Similarly, analysis of 1.7 million apps on the Android market by Juniper's Mobile Threat Center also found that many apps solicit personal information or perform functions not required for the apps to work. For instance, the study found that 94% of free gambling apps that have permission to make outbound calls do not describe why the app would justifiably use this capability. Equally, 83.88% of free gambling apps have permission to use the camera and 84.51% have permission to send SMS messages; a complete lack of transparency as to who is collecting information and how it is used. The study found that other permissions being requested from applications include the ability to initiate outgoing calls, send SMS messages and use a device camera without the user's knowledge. Conversely, access to the device camera could enable a third party to obtain video and pictures of the area where the device is present, he said.

Silently sending SMS messages can also be a means to create a covert channel for siphoning sensitive information from a device.

"Further, the potential for stealth SMS messages or calls can have monetary repercussions by communicating with services that will subsequently charge a fee, such as sending premium SMS messages," said Hoffman.

Research firm Gartner predicts that the number of mobile applications downloaded this year will double to 45 billion.

In the light of this prediction, Hoffman said more needs to be done to inform people about the information being captured, particularly as an increasing number of people use personal devices in the workplace to access business-critical information.

The problem, he said, is that the companies, consumers and government employees who install these apps often do not understand with whom they are sharing personal information.

“Even though a list of permissions is presented when installing an app, most people do not understand what they are agreeing to or have the proper information needed to make educated decisions about which apps to trust,” said Hoffman.

On the other hand, mobile software is a competitive business that changes rapidly and responds to changes in markets, hardware, and software platforms. New projects are born and aged ones are rewritten creating a challenge for vulnerability detection models that rely on historical vulnerability data to detect and/or predict future vulnerability proneness. Because many new projects do not have enough historical data to train detection models, we can use models estimated from any training data available on other projects. One critical reason for the insecurity of mobile Apps is the fact that most developers don’t have the appropriate knowledge about secure coding (Medeiros, Neves, & Correia, 2014). Therefore, App owners should strive to create applications that satisfy all user expectations regarding safety, an instance that is not always the case (SteelKiwi Inc., 2018). Additionally, researchers from University College London (UCL) claim that it is possible for an app running on mobile device to listen and share vital information about the user, even when the app itself is closed. They asserts that worse than market data gathering, the apps can listen for ultrasound signals and could potentially continuously monitor conversations and even record keystrokes since the technology works across devices (Brown, 2016). The UCL researchers say that software that lacks opt-out options and disclosure poses a significant threat.

Software Security Vulnerability Detection (SSVD) is a research field that utilizes effective methods for detecting the vulnerability in a given mobile App component. These methods help security tester allocate their limited resources to the most vulnerable systems. The process of

building secure Apps is expensive, difficult, and time consuming. Building and distributing vulnerability detection models can cause quality assurance teams to focus their time and resources on the vulnerable parts of their code base. Researchers on the other hand usually build vulnerability detection models that use metrics and vulnerability data. Known as unsupervised learning approaches in machine learning fields, these models implement different types of learning algorithms. Finding and solving vulnerabilities in the early stages of mobile Apps composition is an important step. Software vulnerability detection is a quality assurance technique that includes inspection and testing within the Software Quality Engineering discipline (Zhang, Caragea, & Ou, 2011). However, such quality assurance is best done by engineers who are specially trained in software security (McGraw, 2006). Methods and techniques that identify components that are more likely to contain vulnerabilities can provide significant aid to the security engineers who focus their attention on higher risk components. The security of most systems and networks depends on the security of the software running on them. Most of the attacks on these systems occur because of exploiting vulnerabilities found in these software applications (Walden, Doyle, Lenhof, & Murray, 2010). Security failures in software are common and growing and are caused by a vulnerability that can be exploited (Chowdhury & Zulkernine, 2011). It is very challenging to identify vulnerabilities before they manifest themselves as security failures while the software is operating, because security concerns are usually not sufficiently known at early stages of the software development life cycle. Therefore, it is essential to know in advance the characteristics of software files that can indicate vulnerabilities to help software security testers and managers take proactive action against potential vulnerabilities. Security testing is an important requirement of software security even if they are very resource-intensive. The most worrying class of these faults are those that can be exploited recurrently, causing companies to struggle allocating resources for their management (Nyanchama, 2005).

Within-project vulnerability, detection is built from a part of a project and evaluated on the remainder of the project. Cross-project vulnerability detection is done when new projects don't have enough vulnerability data to build a detection model (Abunad & Alenezi, 2015). In this case, we use data from other projects to build a detection model. Several companies and projects might not yet have attained historical information about vulnerabilities in order to build detection models. As a result, the ability of several metrics collected from software are evaluated in this work. These files are used to detect vulnerabilities in a project by using other projects datasets (cross project). The open source community

has responded to this trend with a host of new projects, including solutions that help enterprises track and manage mobile devices, mobile development tools for creating new apps and open source apps that enable greater productivity. In this paper, the vulnerability detection of three open source mobile applications have been studied through a thorough empirical study.

This study employs a data mining technique to discover potentially useful patterns from large data sets generated due to the proliferation of mobile internet of things and applying appropriate algorithms to extract hidden information. Clustering, a data mining technique is used to group data with respect to their similarities, primarily concerned with distance measures between them (Chen, et al., 2015). The technique is adopted for this study due to its ability to analyze data objects without consulting a known class model. According to (Jain & Dubes, 1988), clustering algorithms divide data into meaningful groups so that patterns in the same group are similar in some sense and patterns in different group are dissimilar in the same sense. Searching for clusters involves unsupervised learning (Ansari, et al., 2013). In information retrieval, for example, the search engine clusters billions of web pages into different groups, such as news, reviews, videos, and audios.

2. Literature Review

Abunad and Alenezi (2015) formulated a vulnerability prediction as a classification problem by predicting if a PHP file is vulnerable or not. In their study, they presented a framework for how cross project vulnerability prediction can be used to automatically predict vulnerable files in new projects. On the other hand, (Al-Subaihin, Sarro, Black, & Capra, 2016) presented an app clustering solution consisting of three main stages: 1) feature extraction from the textual description of the mobile apps, 2) feature clustering to reduce the granularity of the features used to describe each app, and 3) clustering conducted over the apps themselves. Other works have been done on clustering and data mining that includes: Jianliang, Haikun , and Ling, (2009) who introduced the application on intrusion detection based on K-means clustering algorithm, to detect unknown attack and partition large data space effectively. This method has many disadvantages though, degeneracy and cluster dependence. Guan, Ghorbani, and Belacel (2003) introduced Y-means algorithm which is a clustering method of intrusion detection, based on K-means algorithm and other related clustering algorithms. It overcomes two short comings of K-means, number of cluster dependency and degeneracy. Mingqiang and WangQian (2012) introduced a new concept of a graph based clustering algorithm for anomaly based clustering algorithm for anomaly intrusion

detection. They used outlier detection method which is based on local deviation coefficient (LDCGB). Compared to other intrusion detection algorithm of clustering this algorithm is unnecessary to initial cluster number. Chitrakar and Chuanhe (2012) proposed a hybrid learning approach of combining k-medoids clustering and naive bayes classification. Because of the fact that k-medoids technique represents the real world scenario of data distribution the proposed algorithm grouped the whole data into clusters more accurately than K-means, resulting in better classification. Jian, (2009) proposed an improved intrusion detection model based on DBSCAN which illustrates the idea of generating a cluster using density method and merging small clusters. The paper describes a more reasonable density-based clustering algorithm for intrusion detection (IIDGB), using rational method to calculate the distance and design the method of parameter selection.

Xue-yong and Guo (2010) proposed a new intrusion detection based on improved DBSCAN which uses an improved density based cluster algorithm to improve the drawbacks of earlier approach as proposed (Jian, 2009) using more rational method for calculating the distance and process of merging cluster. Some of the drawbacks which are overcome are high false alarm rate and generation of small clusters after the experiment. Li and Fang-Cheng Shen (2010) introduced a novel rule-based intrusion detection system using data mining. They proposed an improvement over apriori algorithm by bringing the concept of length-decreasing support to detect intrusion. Association rules and sequence rules are the main technique of data mining. Muda, Yassin, Sulaiman, and Udzir (2011) published new work of Intrusion detection based on K-means clustering and OneR classification; they proposed an approach which combines the techniques of K-means and OneR classification with the main goal being to utilize K-means algorithm and to split and group data into normal and attack instances. The algorithm partitioned the dataset into k clusters according to an initial value known as seed point into each cluster's centroids or cluster centers and mean value of each data contained within each cluster called centroids. (Li, Li, and Xu (2011) proposed anomaly intrusion detection method based on K-means clustering algorithm with particle swarm optimization. Particle swarm optimization (PSO) algorithm is an evolutionary computing technology based on swarm intelligence and has a good global search ability. The proposed algorithm has overcome falling into local minima and has relatively good overall convergence.

Wankhade and Patka (2013) gave an overview of intrusion detection which is based on data mining techniques. They discussed the various data mining techniques which can be applied on intrusion detection system for the effective identification of both known and unknown pattern of attack in order to develop a secure information system. Fatma and Mohamed (2013) proposed a two stage technique to improve intrusion detection system based on data mining algorithm. They adopted a two stage technique in order to improve the accuracy of sensors. The first stage aim to generate meta-alerts through clustering and the second stage aims to reduce the rate of false alarms using a binary classification of the generated meta-alerts. For the first stage they used two alternatives, self-organizing map (SOM) with K-means algorithm and neural GAS with fuzzy cmeans algorithm and for the second stage they used three approaches, SOM with K-means algorithm, support vector machine and decision trees. Chandrasekhar and Raghuveer (2013) introduced a new concept of Intrusion detection technique by using K-means, fuzzy neural network and SVM classifiers. The proposed technique has four major steps: first one is to use Kmeans algorithm to generate different training subsets. Based on the obtained training subsets, different neuro-fuzzy models are trained. Then a vector for SVM classification is formed and lastly, classification using radial SVM is performed to detect if intrusion has occurred or not.

The different approaches discussed above are proposed for intrusion detection using various techniques of data mining. The techniques have many advantages over earlier approaches but they are not good in all respect. In this research, we present a new approach for anomaly intrusion detection by using new medoid algorithm of K-medoid and some modifications of it. The rest of the study is organized as follows: subsection 2.1 describes the existing K-means algorithm in brief, subsection 2.2 describes our proposed work in detail, section 3 describes the experimental design and results for the proposed work, section 4, execution and discussion and finally in the last section 5, we discuss conclusion and future work.

2.1. K-MEANS ALGORITHM

For the completeness of the paper we discuss the K-means algorithm in this section. K-means is an iterative clustering algorithm in which items are processed among set of clusters until the desired set is reached. K-means algorithm is like a squared error algorithm. Using K-means algorithm we achieve a high degree of similarity among elements and dissimilarity in different clusters.

Given a cluster K_i , let the set of items mapped to that cluster be $\{t_{i1}, t_{i2}, t_{im}\}$.

The cluster mean of $K_i = \{t_{i1}, t_{i2}, t_{im}\}$ is defined as

$$m_i = \frac{1}{m}\sum_{j=1}^{m} t_{ij} \quad (1)$$

This definition indicates each tuple has only one numeric value as opposed to a tuple with many attribute values. The following sum up the steps of K-means algorithm (Dunham,, 2003)[17].

K-means Algorithm (Ranjan & Sahoo , 2014)

Input: E= $\{t_1, t_2 \ldots t_n\}$ c //number of desired clusters

Output: C //Set of clusters

Begin

Assign initial values for mean m_1, m2…m_k;

Repeat

Assign each item t_i to the cluster 'c' which has the closest mean. Calculate new means for each cluster.

Until

Convergence criteria are met.

End

K-means has been widely used algorithm however it has many disadvantages. It includes dependence on initial centroids, dependence on number of clusters and degeneracy. To overcome these disadvantages we have proposed a new algorithm.

2.2. Proposed Approach

The approach in this study is and improvement of works by Chitrakar and Chuanhe (2012) of combining k-medoids clustering and naive bayes classification by introducing some modification in representing real world scenario of data distribution in grouping the whole data into clusters for better classification. After running the model on training dataset, we test this model on a third project and evaluate its effectiveness in detecting vulnerabilities of another project (cross project detection).

Cross-project vulnerability detection models are trained on data from one or more projects for which detectors and actual vulnerabilities are available. Then, machine learning techniques are used to build detection models to detect the vulnerabilities mobile application files of a new project. Section 4 gives detailed description of our approach.

3. Experimental Design

The following points discuss about the steps of an experiment conducted.

3.1. Input Data Set

This study used data originally used by (Meidan, et al., 2017) obtained from University College London (UCL) repository aimed at distinguishing between benign and malicious traffic data by means of anomaly detection techniques. In their experiment, malicious data can be divided into 10 attacks carried by 2 botnets, the dataset can also be used for multi-class classification: 10 classes of attacks, plus 1 class of 'benign'. Their study results were that for each of the 9 IoT devices trained and optimized a deep autoencoder on 2/3 of its benign data. This was done to capture normal network traffic patterns. The test data of each device comprised of the remaining 1/3 of benign data plus all the malicious data. On each test set they applied the respective trained (deep) autoencoder as an anomaly detector. The detection of anomalies (i.e., the cyber-attacks launched from each of the above IoT devices) concluded with 100% TPR (UCL, 2018).

3.2. Data Standardization

Dataset standardization process was done to be usable by the proposed algorithm using the following steps:

Step 1: Find the mean of each feature in the dataset using the equation

$$\text{Mean}_f = \frac{D_{1f} + D_{2f} + \cdots D_{nf}}{n} \quad (2)$$

Where, D_{1f}, $D_{2f \ldots Dnf}$ are n measuring values of each feature f.

Step 2: Compute the standard deviation of the calculated mean using the equation

$$SD_f = \frac{1}{n}(|D_{1f} - Mean_f| + |D_{2f} - Mean_f| + \ldots |D_{nf} - Mean_f|) \quad (3)$$

Where, D_{1f}, D_{2f}...D_{nf} are n measuring values of each feature f.

Step 3: The standardized value are given as:

$$S_{if} = \frac{D_{if} - Mean_f}{SD_f} \quad (4)$$

4. Execution and discussion

After getting the standardized value the proposed algorithm is being run to get the desired cluster and later we choose the cluster to label them as normal or intrusions. The proposed algorithm is compared against the existing algorithms on the basis of parameters detection rates, accuracy and false alarm rate. The different metrics used for the purpose of checking performance and observing experimental results are given as under: 1) Detection Rate (DR), the number of intrusion instance detected by the system (True Positive) divided by the total number of intrusion instances present in the test dataset; 2) False Alarm Rate (FAR), the number of normal patterns classified as attacks (False Positive) divided by total number of normal patterns; 3) False Positive (FP), corresponds to an event signaling IDS to produce an alarm when no attack has taken place; 4) False Negative (FN), a failure of IDS to detect an actual attack; 4) True Positive (TP), corresponds to a legitimate attack which triggers an IDS to produce an alarm; and 5) True Negative (TN), corresponds to a situation when no attack has taken place and no alarm is raised.

Therefore, the Detection rate is measured using the following formula:

Total no of Intrusion Instance= TP+FP

$$\text{Detection Rate (DR)} = \frac{TP}{\text{TOTAL NO OF INTRUSION INSTANCE IN DATASET}} \quad (5)$$

Accuracy is measured by the following formula:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

False alarm rate is given by the following formula:

$$\text{False Alarm Rate (FAR)} = \frac{FP}{FP+TN} \quad (7)$$

The Detection Rate, Accuracy and False Alarm Rate are calculated to check the performance of our proposed approach with other existing algorithms. The figure 1 shows the percentage of accuracy for different algorithms including proposed approach. The accuracy of the proposed algorithm is 96.38% which is moderately higher than the existing K-means, FCM, and Y-means algorithms.

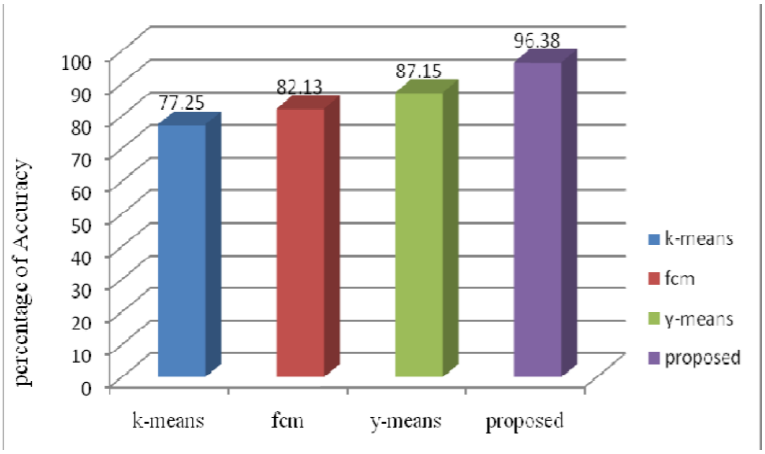

Figure 1 Accuracy comparison

Table 1: Detection Rate of Proposed Algorithm, K-means, FCM and Y-means

Parameters	**K-means**	**FCM**	**Y-means**	**Proposed algorithm**
Detection Rate	82.3	84.6	86.3	91.2
Accuracy	77.25	82.13	87.15	96.38
False Alarm Rate	5.2	4.2	3.9	3.2

The Table 1 gives the percentage of detection rate, accuracy and false alarm rate. The detection rate for the proposed method is 91.2 which is greater than K-means (82.3), Y-means (86.3) and FCM (84.6). Similarly the accuracy for our proposed algorithm is 96.38 and is greater than K-means (77.25), Y-means (87.15) and FCM (82.13). Also false alarm rate is 3.2 which is lesser than that of K-means (5.2), Y-means (3.9) and FCM (4.2). The Table 2 shows the percentage of various attacks including denial of service attack percentage, remote to local attack percentage, user to root attack percentage and probe attack percentage which are used as a measuring parameter to check the percentage of accuracy and detection rate of our proposed algorithm and various existing algorithms.

Table 2.Experiment Results of Proposed Algorithm, K-means, FCM, and Y-means

Parameters	**K-means**	**FCM**	**Y-means**	**Proposed algorithm**
Denial of service (%)	79.83	83.12	89.15	96.12
Remote to local (%)	78.12	82.45	85.10	90.10
User to root (%)	52.10	60.10	65.12	70.51
Probe (%)	62.45	65.25	68.12	70.13

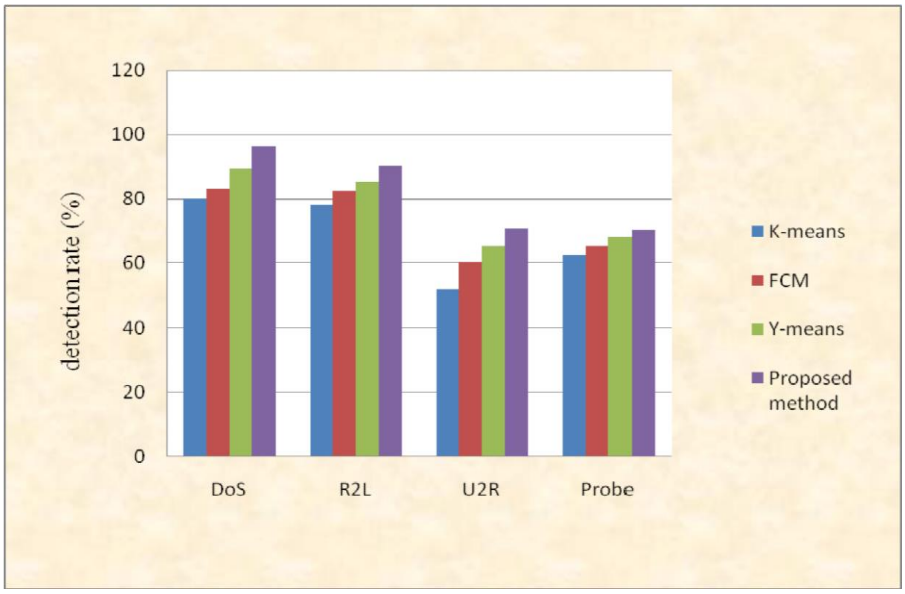

Figure 2. Comparisons of Detection Rate against various attacks

The Figure 2 illustrates the percentage of detection rate of denial of service attack, remote to local attack, user to root attack and probe attack and comparisons are made for proposed method, K-means algorithm, Y-means algorithm and FCM algorithm. We observed from the figure that detection rate for our proposed approach is greater than earlier approaches. The result findings show that the proposed algorithm is able to detect greater number of intrusions with higher accuracy compared to existing algorithm. The detection rate of Denial of service attack and Remote to local attack is greater than user to Root attack and Probe attack. Results also show that there are lesser false positive rate and false negative rate. In order to improve detection rate and accuracy a modified version of k-medoid algorithm is used which eliminates the disadvantages of K-means algorithm. Also the degeneracy is being eliminated by using the concept of searching for the empty clusters and deleting them. The initial medoid selection is the important task for the algorithm since it affects the performance of the proposed algorithm. By using an efficient method for the selection of initial medoid the algorithm performs faster execution and overcomes the disadvantages of existing algorithm.

5. Conclusion and Future Work

The approach proposed in this study describes the new way of intrusion detection using k- medoid clustering algorithm and some modifications of it; specifying a new way to select initial medoid and proved to be better than K-means for anomaly intrusion detection. The algorithm conveys the idea of data mining technologies which is certainly a good field and popular area of research in intrusion detection. The proposed approach is having many advantages over the existing algorithm which mainly overcomes the disadvantages of dependency on initial centroids, dependency on the number of cluster and irrelevant clusters. The algorithm is able to sort out these problems and has been able to provide high detection rates and less false negative rate. Nonetheless, the algorithm has few disadvantages which need to be focused in future studies; including detection rate for probe and user to root attack.

References

Abunad, I., & Alenezi, M. (2015). Towards Cross Project Vulnerability Prediction in Open Source Web Applications. *ICEMIS '15.* Istanbul, Turkey.

Li, L., & Fang-Cheng Shen, D.-Z. (2010). A novel rule-based Intrusion Detection System using data mining. *In ICCSIT, IEEE International conference,.*

Li, Z., Li, Y., & Xu, L. (2011). Anomaly intrusion detection method based on K-means clustering algorithm with particle swarm optimization. *In ICM.*

Al-Subaihin, A., Sarro, F., Black, S., & Capra, L. (2016). Clustering Mobile Apps Based on Mined Textual Feature.

Ansari, S., Chetlur, S., Prabhu, S., Kini, G., Hegde, C., & Hyder, Y. (2013). An overview of clustering analysis techniques used in data mining. *International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 12*, 284–286.

Brown, B. (2016). *Some mobile apps continue to track ultrasound signals even when closed*. Retrieved from finance.yahoo.com: https://finance.yahoo.com/news/computer-apps-may-keep-listening-224643121.html?guccounter=1

Chandrasekhar, A., & Raghuveer, K. (2013). Intrusion detection technique by using K-means,fuzzy neural network and SVM classifiers. *In ICCCI.*

Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A., & Rong, X. (2015). Data Mining for the Internet of Things:Literature Review and Challenges. *International Journal of Distributed Sensor Networks*.

Chitrakar , R., & Chuanhe, H. (2012). Anomaly detection using Support Vector Machine Classification with K-Medoids clustering. *In Internet (AH-ICI), 3rd Asian Himalayas International conference,*.

Chowdhury , I., & Zulkernine, M. (2011). Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities. *Journal of Systems Architecture, 57(3)*, 294–313.

Dunham,, M. (2003). *Data Mining: Introductory and Advanced Topics,ISBN: 0130888923,*. published by Pearson Education, Inc.

Fatma, H., & Mohamed, L. (2013). A two-stage technique to improve intrusion detection systems based on data mining algorithms. *In ICMSAO.*

Fenton , N., & Bieman, J. (2014). *Software metrics: a rigorous and practical approach.* . CRC Press.

Guan, Y., Ghorbani, A., & Belacel, N. (2003). *Y-means: a clustering method for Intrusion Detection. In Canadian Conference on Electrical and Computer Engineering.* Montral, Qubec, Canada,.

Harvey, C. (2015). *50 Open Source Mobile Tools*. Retrieved from www.datamation.com: https://www.datamation.com/open-source/50-open-source-mobile-tools-1.html

Jain , A., & Dubes, R. (1988). *Algorithms for Clustering Data,* . Englewood Cliffs, NJ, USA: Prentice Hall.

Jian, Y. (2009.). An Improved Intrusion Detection Algorithm Based on DBSCAN. *Micro Computer Information, 25,1008-0570(2009)01- 3- 0058-03, 58-60,*.

Jianliang, M., Haikun , S., & Ling, B. (2009). The Application on Intrusion Detection based on K-Means Cluster Algorithm. *International Forum on Information Technology and Application.*

Korolov, M. (2018). *Open source software security challenges persist*. Retrieved from www.csoonline.com: https://www.csoonline.com/article/3157377/application-development/open-source-software-security-challenges-persist.html

McGraw, G. (2006). *Software security: building security in, volume 1*. Addison-Wesley Professional.

Medeiros, I., Neves, N., & Correia, M. (2014). Automatic detection and correction of web application vulnerabilities using data mining to predict false positives. *In Proceedings of the 23rd international conference on World wide web* (pp. 63–74). ACM.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., & Elovici, Y. (2017). 'N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing, Special Issue - Securing the IoT*.

Mingqiang, Z., & WangQian , H. (2012). A Graph-based Clustering Algorithm for Anomaly Intrusion Detection. *In computer science and education (ICCSE), 7th International Conference.*

Muda, Z., Yassin, W., Sulaiman, M., & Udzir, N. (2011). Intrusion Detection based on K-Means Clustering and OneR Classification. *In Information Assurance and Security (IAS), 7th International conference.*

Nyanchama, M. (2005). Enterprise vulnerability management and its role in information security management. *Information Systems Security, 14(3),* 29–56.

Ranjan , R., & Sahoo , G. (2014). A NEW CLUSTERING APPROACH FOR ANOMALY INTRUSION DETECTION. *International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.4, No.2*, 29-38.

Song, S. (2011). *Analysis and acceleration of data mining algorithms on high performance reconfigurable computing platforms [Ph.D. thesis]*. Iowa State University.

SteelKiwi Inc. (2018). *Mobile Application Security: Best Practices for App Developers*. Retrieved from hackernoon.com: https://hackernoon.com/mobile-application-security-best-practices-for-app-developers-1a6345750b35

UCL. (2018). *detection_of_IoT_botnet_attacks_N_BaIoT Data Set* . Retrieved from archive.ics.uci.edu: https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

Walden, J., Doyle, M., Lenhof, R., & Murray, J. (2010). Idea: java vs. php: security implications of language choice for web applications. *In Engineering Secure Software and Systems,* (pp. 61–69). Springer.

Wankhade, K., & Patka, S. (2013). Ravindra Thool, An Overview of Intrusion Detection Based on Data Mining Techniques. *In Proceedings of 2013 International Conference on Communication Systems and Network Technologies,* (pp. 626-629). IEEE.

Xue-yong, L., & Guo- , G. (2010). A New Intrusion Detection Method Based on Improved DBSCAN. *In Information Engineering (ICIE), WASE International conference.*

Zhang, S., Caragea, D., & Ou, X. (2011). An empirical study on using the national vulnerability database to predict software vulnerabilities. *In Database and Expert Systems Applications, pages* (pp. 217–231). Springer,.