Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 110 (2017) 230–235

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017)

A novel approach in detecting intrusions using NSLKDD database and MapReduce programming

Amar Meryem*, Douzi Samira, El Ouahidi Bouabid, Lemoudden Mouad

Mohammed-V University, Faculty of Sciences, L.R.I.
B.O. 1014, Rabat, Morocco

Abstract

Due to the increasing usage of the cloud computing architecture, computer systems are facing many security challenges that render sensitive data visible and available to be counterfeited by malicious users and especially intruders. Log files are generated at every level of the computing infrastructure and represent a valuable source of information in detecting attacks. The main goal of this work is the identifiction and prediction of attacks and malicious behaviors by analyzing, classifying and labeling recorded activities in log files. This paper uses MapReduce programming to prior each user behavior, it also employs K-Means algorithm to cluster unknown events and K-NN supervised learning on NSLKDD database to define unlabelled classes.

Keywords: Log files, NSLKDD, K-Means, K-NN, variance-covariance matrix;

1. Motivation

Because of the rapid growth of digitization, the use of cloud computing architecture has become essential in almost all cases[1], it provides availability to end users, resources as needed and without interaction with the service provider[2,3]. However, using the same pool of physical infrastructure by different users and sharing the same network involves the risk of rendering sensitive data visible to other, causing multiple attacks[4, 5, 6]. Moreover, even a trusted person who has authorized access to an application can intentionally cause fraud by deleting functionalities, disabling information accessibility and removing his log file traces.

Considering the importance of log files in recording information about user identification, transaction protocols, and occurring events, we proposed, in our previous work[3, 7, 8, 9], to centralize their storage in secured HDFS (Hadoop

* Corresponding author. Tel.: +212-666-315-707;
E-mail address: amar.meryem@gmail.com

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.06.089

Distributed File System) format. Our proposed storage architecture, a SPOC (Single Point of Contact), firstly upgrades the management of various systems based on the misuse detection, and secondly makes analysis and tracking easier. Finally, the SPOC collects information about the behavior of attackers, especially insiders. After analyzing the cloud-generated log files, we applied a mining algorithm that highlighted correlations between several attacks and predicts the following attack when detecting a succession of detected anomalies.

The results of our previous work[3] show that centralization and log file analysis based on misuse detection is a largely useful approach in cyber security that helps in predicting anomalies. Nevertheless, misuse detection considers each recorded event that is not on a list of known attack signatures as normal and does not calculate the distance between a suspicious behavior and an attack or predict new anomalies altogether. Furthermore, this approach needs a frequent and static update of attack signatures. The FP-Growth algorithm mines frequent attacks, represents them in a tree called FP-tree and gives priority to each detected attack based on its frequency[10]. This mining approach anticipates only with a certain level of reliability by using support and accuracy indicators that calculate the probability of incoming attacks based on previously detected ones. However, it does not predict new attacks or substitute new activities with their correlated ones.

In this paper, we propose an extension and corrections of our previous architecture[3] in five main sections divided as follows. In this first section, we depict our motivation and the limits of our previous solution. The second section reviews the related works as a state of the art and foundation of our study. In the third section, we showed the importance of using NSLKDD as a knowledge database. Then, in the fourth section, we detailed our new architecture in five sub-sections. We conclude this paper by providing a summary and perspectives of our work.

2. Related Works

In the AI² solution Kaylan, Ignacio et al[11] proposed an end-to-end active learning security system based on analyzing log files, applying both a supervised and an unsupervised learning algorithms to label new examples and update the existing intrusion detection rules. The output of the unsupervised learning algorithm is validated by a human analyst. The system then takes into consideration the analyst interactions and updates the labels of the supervised training data. In our case, the human analyst is replaced by the NSLKDD database which must be updated by the results of previous analysis.

Manjula Belavagi and Balachandra used a predictive model[12] to decide if a network traffic is normal or malicious. In their study, they used a number of classification algorithms: Logistic Regression, Gaussian Naïve Bayes, Support Vector Machine and Random Forest on the same dataset. The results were tested by the NSLKDD dataset and showed that Random Forest performs better than the others.

Rana Aamir Raza and Xi-Zhao et. Al[13] worked on reducing the number of intruder polymorphic mechanisms in masquerading the same attack payload to escape security measures. Considering that obtaining a sufficient labeled data for a supervised learning algorithm is cumbersome, they decided to use a semi-supervised fuzziness learning algorithm by relating unlabeled samples to the labeled. The experiment used the NSLKDD dataset and proved that their classifier surpasses the performance of Naïve Bayes, Support Vector Machine and Random Forests algorithms.

3. NSLKDD database

NSLKDD solves some omissions of KDDCUP'99 [14,15,16, 17] and improved the accuracy of the machine learning by reducing the number of biased classifiers.

In User to Root (U2R), the attacker accesses a normal user account by sniffing passwords. These types of attacks are noticed by the NSLKDD features "number of file creation" and the "number of shell prompts invoked". In Denial of Service (DOS) attacks, malicious users fill up the memory resource and the revealing features are "source bytes" and the "percentage of packets with errors". Probes attacks are based on gaining the information of a remote victim by a port scanning and uses "duration of connection" and "source bytes" NSLKDD attributes. Remote to user attacks occur when an attacker who has the ability to send packets to a machine and does not have an account on that machine exploits some vulnerabilities to gain local access. This type of attack uses mainly "duration of connection", "service request" and "number of failed login attempts"[16, 18].

4. Proposed Architecture

The object of our architecture is to identify attacks by labeling recorded user behaviors. In order to make simpler the detection of intrusions from a massive set of logs we started by mapping each logged line in the SPOC into a row matrix and applied MapReduce algorithm on it to mine frequent behaviors. Then we apply a machine learning algorithms to classify stored activities, label detected attacks and predict new malicious behaviors. The figure bellow fig.1 demonstrates our solution in five steps as follows:

- Step 1: Log Centralization and Matrix Representation
- Step 2: Affecting event weights by MapReduce Programming
- Step 3: Classifying unlabeled behaviors with k-means
- Step 4: Labeling Behaviors using NSLKDD training dataset
- Step 5: Finding correlations between features

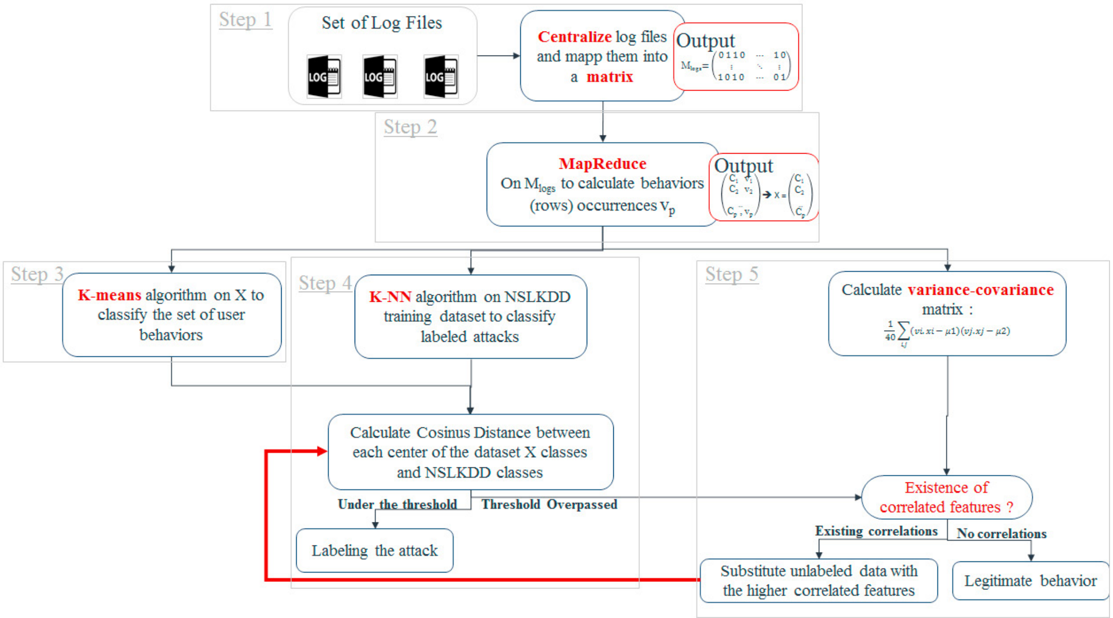
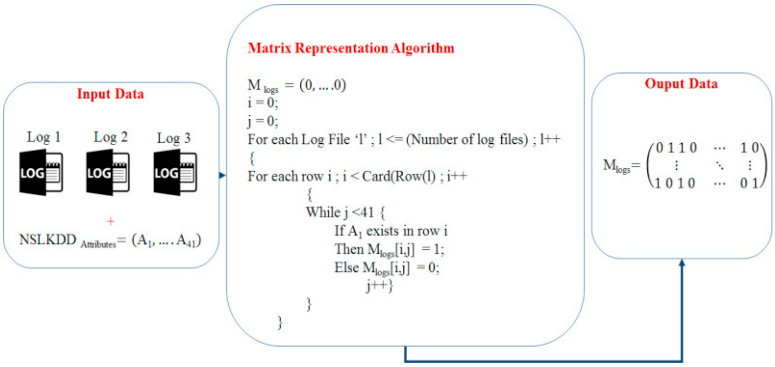

Step 1
Set of Log Files
LOG
LOG
LOG
Centralize log files and mapp them into a matrix
Output
$M_{logs}=\begin{pmatrix}0110 & \cdots & 10\\ \vdots & \ddots & \vdots\\ 1010 & \cdots & 01\end{pmatrix}$
Step 2
MapReduce
On M_{logs} to calculate behaviors (rows) occurrences V_p
Output
$\begin{pmatrix}C_1 & v_1\\ C_2 & v_2\\ C_p & v_p\end{pmatrix} \rightarrow X=\begin{pmatrix}C_1\\ C_2\\ C_z\end{pmatrix}$
Step 3
K-means algorithm on X to classify the set of user behaviors
Step 4
K-NN algorithm on NSLKDD training dataset to classify labeled attacks
Calculate Cosinus Distance between each center of the dataset X classes and NSLKDD classes
Under the threshold
Threshold Overpassed
Labeling the attack
Step 5
Calculate variance-covariance matrix :
$\frac{1}{40}\sum_{i,j}(vi.xi-\mu1)(vj.xj-\mu2)$
Existence of correlated features ?
Existing correlations
No correlations
Substitute unlabeled data with the higher correlated features
Legitimate behavior

Fig. 1. Proposed Architecture

4.1. Step 1: Log Centralization and Matrix Representation

In order to extract sensitive information and identify vulnerabilities and correlations[4] between user behaviours, we follow up on our previous work.


Input Data
Log 1
Log 2
Log 3
LOG
LOG
LOG
+
NSLKDD $_{Attributes}$= $(A_1, ... A_{41})$
Matrix Representation Algorithm
Ouput Data
$M_{logs}=\begin{pmatrix}0\,1\,1\,0 & \cdots & 1\,0\\ \vdots & \ddots & \vdots\\ 1\,0\,1\,0 & \cdots & 0\,1\end{pmatrix}$

```
M logs = (0, ….0)
i = 0;
j = 0;
For each Log File ‘l’ ; l <= (Number of log files) ; l++
{
For each row i ; i < Card(Row(l) ; i++
          {
          While j <41 {
               If A1 exists in row i
               Then Mlogs[i,j] = 1;
               Else Mlogs[i,j] = 0;
                    j++}
          }
     }
```

Fig. 2. Mapping log files into $\mathbf{M_{logs}}$ matrix

We firstly centralize all generated logs in the SPOC and then each line in a log file will be mapped to the NSLKDD vector having 41 elements that refers respectively to the 41 NSLKDD features[16]. For each record (**R_Rows**) in a selected log file we verify the existence of each feature of the NSLKDD and write 1 if the attribute exists and 0 if not and insert these vectors in a 41×N matrix named **M_{logs}**, with N the sum of relevant rows of each log file. This projection reduces extremely the volume of the input dataset and facilitates the training phase process.

4.2. Step 2: Affecting event weights by MapReduce algorithm

Considering the increasing scale of log file volume and velocity[4, 8], the resulting matrix from the previous step amounts to a similar size. Given the size of this matrix, traditional algorithms would be inadequate to compute frequencies and extract existing correlations between NSLKDD attributes. In this step, we will be using a MapReduce program[9] that takes in the **M_{logs}** matrix as input and calculates the number of occurrences of each NSLKDD feature. The fig. 3 bellow describes the MapReduce program.

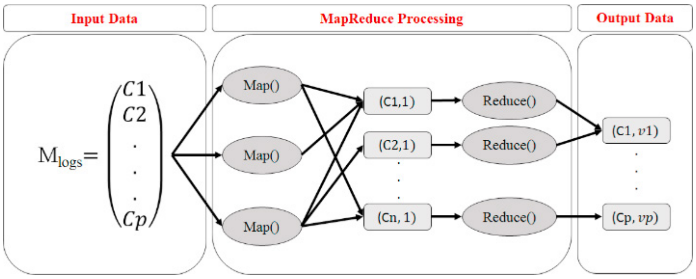


Fig. 3. MapReduce Processing

The mapper attributes a unique identifier to each row of the matrix and initialize the occurrences to 1. It is important to note that each distinct row signifies a distinct behavior present in the log file. In this sense, we are trying to calculate the number of occurrences of each distinctive behavior in the log file. The reducer calculates the occurrences of each row in the matrix. The output of the previous processing is a list of **(C_i, v_i)** couples, where **C_i** is the unique identifier of a distinct row and v_i is its occurrence. After that, each frequency is considered as a weight of each **C_i**. As a result, **M_{logs}** is reduced into a 41×P matrix **X** with $P \leq N$, and P is the count of distinct row behavior.

$$N = \sum_{i=1}^{p} v_i \qquad (1)$$

4.3. Step 3: Classifying unlabeled behaviors with k-means

In this phase, we classify unlabeled behaviors of the reducer processing output data using k-means algorithm[19, 20]. k-means is an unsupervised classifier that does not require a correct or labeled answer associated to each input pattern in the training data. but explores the underlying and hidden structure in the data, revealing correlations between patterns and organizing them into several categories based on the 41 NSLKDD features. Classifying the unknown behaviors groups events into classes of similar characteristics and, applying k-means classifier on **X**, partitions our reduced matrix into k clusters having related attributes. Thus, finding correlated attributes returns to get the argument x_i that converges to the mean value of the column **X_j**. Then, x_i must verify the following formula (3).

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{x \in c_i} \left\| v_i . x_i - \mu_j \right\|^2 \qquad (2)$$

Where, **C_i** is the i^th class and **C** a vector of k classes and μ_j the mean value of the j^th column of **X**

4.4. Step 4: Labeling Behaviors using NSLKDD training dataset

Labeling unknown behavior’s phase starts with classifying the labeled attacks and signatures of malicious behaviors. In this first phase, we consider NSLKDD training dataset as a knowledge database of attacks and apply the k-nearest neighbor classifier on it to output five classes membership.

Applying KNN algorithm to our NSLKDD labeled training dataset helps us define the attributes of each common known attack and find correlations between the features[21, 22]. Then, the result of the KNN algorithm validates the results of the previous unsupervised process by calculating the 'cosine' distances between each NSLKDD center class and K-means output classes. If the distance does not overpass a threshold this means that the two compared classes are close to each other and we label the unknown class with its correlated to a certain probability. If not the unlabeled class is communicated to the following processing (step 5).

4.5. Step 5: Finding correlations between features

Calculating the variance-covariance matrix uncovers the relationship[23] between the NSLKDD attributes, enables the system to discover new malicious behaviors and predict new signatures in identifying an attack. Based on the matrix representation of input data projected on NSLKDD features, we observe 41 independently distributed N-dimensional vectors $\mathbf{X_i}$, i Є {1,…,41}. In order to identify correlations between features and indicate whether variables are positively or inversely related, we calculate the covariance of the Matrix **X**. with *N*>41 and estimate the covariance matrix Σ from the data **X = (X₁, …,X₄₁),** which results to a 41×41 matrix that we proposed in the formula (4).

$$\Sigma = \begin{bmatrix} \mathrm{var}(x_1) & \dots & \mathrm{cov}(x_1, x_{41}) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(x_{41}, x_1) & \cdots & \mathrm{var}(x_{41}) \end{bmatrix} \tag{3}$$

Var($\mathbf{X_1}$) represents the variance of $\mathbf{X_1}$ which is the expected value of the squared deviation from the mean of **X**, $\mu(X_1)$ and Cov(x_1, x_2) represents the covariance of x_1 and x_2 :

$$Var(X_1) = \frac{1}{40} \sum (v_i.x_i - \mu)^2 \tag{4}$$

$$Cov(X_1, X_2) = \frac{1}{40} \sum_{i,j} (v_i.x_i - \mu_1)(v_j.x_j - \mu_2) \tag{5}$$

In this step, correlations[14, 24] between the attributes permit the processing predicting new behavior and attributes in detecting an attack. Then, the rest of unlabelled data in the previous step are substituted with their highly-correlated features and redirected to the previous process (Step 4) where cosine distances are calculated to label substituted attributes. The fourth and fifth steps are repeated until the results converges. Finally, the residual unlabeled behaviors are considered as legitimates.

5. Perspectives and conclusions

In this study, we centralized the generated log files to disable intruders from deleting their traces and improve cloud security by enabling the possibility of finding correlations between user behavior and predicting new attacks. In the SPOC we applied a MapReduce program to firstly handle the big volume and velocity of generated log files and assign a weight to each recorded activity. After that, we improved the labeling of unknown classified behaviors by calculating cosine distances between the behavior classes and the NSLKDD training dataset classes. Considering the fact that using only this approach in labeling event is insufficient, we introduced the results of variance-covariance matrix to substitute each attribute with its highly correlated one.

Our proposed methodology allows us to classify detected attacks during the training phase and subsequently allows us to know, with a certain probability, if an unknown behavior is associated with the existence of an attack. In order to prove the exactitude of our solution, we are presently implementing this approach using the Python programming language and scikit-learn libraries on the HDFS Hadoop architecture.

References

1. Adetunmbi A., Adeola , Dramola O. «Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features,» *Engineering and Computer Science,* vol. I, pp. 1-7, 2010.

2. Targio I., Ibrar Y., Nor Badrul A. et al « The rise of "big data" on cloud computing: Review and open research issues,» *Information Systems,* pp. 98-115, 2015.
3. Grance T., Peter M. «The NIST Definition of Cloud Computing,» *NIST Special Publication,* 2011.
4. Amar M., Lemoudden. M. El. Ouahidi. B. «Log File's Centralization to Improve Cloud Security,» *IEEE Xplore,* 2017.
5. Mazhar A., Samee U., Athanasios V. «Security in cloud computing: Opportunities and challenges,» *Information Sciences,* pp. 357-383, 2015.
6. Amruta A., Narendra S., «Insider threat Detection using Log analysis and Event Correlation,» *Procedia Computer Science,* pp. 436-445, 2015.
7. Modi C., Dhiren P., Bhavesh B., Hiren P., Muttukrishnan R. «A survey of intrusion detection techniques in cloud,» *Journal of Network and Computer Applications,*pp. 42-57, 2012.
8. Lemoudden M.,El ouahidi B. «Managing Cloud-generated Logs Using Big Data Technologies,» *IEEE Xplore*, 2015.
9. Lemoudden M., Amar M., El Ouahidi B. «A Binary-Based MapReduce Analysis for Cloud Logs,» *Procedia Computer Science,* pp.1213-1218*,* 2016.
10. Ahmad R., Osman Fageeri S. «An Efficient Log File Analysis Algorithm Using Binary-Based Data Structure,» *Procedia Social and Behavioral Sciences,* pp. 518-526, 2013.
11. Prabhakar R., Ramraj T., «Frequent Subgraph Mining Algorithms –A survey,» *Procedia Computer Science,* pp. 197-204, 2015.
12. Kalyan. V., Ignacio. A., «AI^2: Training a big data machine to defend,» *IEEE Xplore,* pp. 1-13, 2016.
13. Manjula. C., Balachandra. M., «Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection,» *Procedia Computer Science,* pp. 117-123, 2016.
14. Rana A.,Xi.-Zhao. W., Joshua Z. «Fuziness based semi-supervised learning approach for intrusion detection system,» *Information Sciences,* pp. 484-497, 2017.
15. Bolón-Canedo V.,Sanchez.-M.,Alonso B. «Feature selection and classification in multiple class datasets: An application to KDD,» *Expert Systems with Applications,* pp. 5947-5957, 2011.
16. Dhanabal L., Shantharajah. S. «A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms,» *International Journal of Advanced Research in Computer and Communication Engineering,* pp. 1848-1853, 2015.
17. Preeti A.,Sudhir. K. S. «Analysis of KDD Dataset Attributes - Class wise For Intrusion Detection -,» *Procedia Computer Science,* p. 842 – 851, 2015.
18. Chae H., Byung-oh. J. Sang-hyun. C. «Feature Selection for Intrusion Detection using NSL-KDD,» *Recent advances in computer science,* pp. 361-763, 2013.
19. Marco C.,Aritz. P. , Jose. A. «An efficient approximation to the K -means clustering for massive data,» *Knowle dge-Base d Systems,* pp. 56-69, 2016.
20. Zhaohui J., Tingting. L. ,. Wenfang. M. , Zhao. Q. ,. Yuan. R., «Fuzzy c-means clustering based on weights and gene expression programming,» *Pattern Recognition Letters,* pp. 1-7, 2017.
21. Ming-Yang Su., «Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification,» *Journal of Network and Computer Applications,* pp. 722-730, 2011.
22. Yali W., Brahim. C. «KNN-Based Kalman Filter: An Efficient and Non-stationary Method for Gaussian Process Regression,» *Knowledge-Based Systems,* pp. 148-155, 2016.
23. Wessel. N. van Wieringen «On the mean squared error of the ridge estimator of the covariance and precision matrix,» *Statistics and Probability Letters,* pp. 88-92, 2017.
24. Samprit B., Stefano. M, Martin. T. W. «A regularized profile likelihood approach to covariance,» *Journal of Statistical Planning and Inference,* vol. 179, pp. 36-59, 2016.