Volume 3 No. 9, Sep-Oct 2013 ISSN 2223-4985

International Journal of Information and Communication Technology Research

http://www.esjournals.org

Privacy Preserving Association Rule Mining in Collaborative Intrusion Detection Systems with Fuzzy Data

Motahareh Dehghan Chachkamy, Babak Sadeghiyan
Data Security Lab, Amirkabir University of Technology,
424 Hafez Ave, Tehran, Iran

ABSTRACT

One area of research in information security is Intrusion Detection Systems (IDSs), which are installed on target systems and tracks the indication of attacks. Since, Intrusion Detection Systems for attack detection, need to information collection and analysis, concerns about the disclosure of individuals and systems and/or disclosure sensitive information of them exist. Therefore, despite the use of IDSs, we deal with a privacy violation.

One of the issues of privacy in network intrusion detection systems (NIDSs) is that several organizations wish to collaborate together to prevent the penetration of their sites. To achieve this, they share normal and attack data of their IDSs. Since the data are sensitive, they don’t want to share explicit data. Now, how these organizations can operate data mining and/ or machine learning on aggregate data without violation on data confidentiality. Privacy concerns can prevent this approach - there may not be a central site with authority to see all the data. We present a privacy preserving algorithm to mine association rules from several organizations (IDSs). These organizations, partitioned horizontally. In this paper, we describe weighted Association Rule Mining from fuzzy and binary data, using secure sum method. This paper generally focuses on the association rule mining from KDD dataset and for instance, generates Neptune attack rules that will detect Neptune attack in network audit data using anomaly detection.

Keywords: *privacy, Association Rule (AR), Collaborative Intrusion Detection System (IDS), KDD Dataset, Secure Sum, weighted support and confidence.*

1. INTRODUCTION

Today, with the high-speed development of computer networks, a network-based computer system plays increasingly vital roles in all kinds of branches. This has made information security a more and more important problem. Many techniques have been developed to insure information security, such as virus prevention, firewall, safe router, etc. Since it is not technically feasible to build a system with no vulnerabilities, intrusion detection has become an important area of research .also, Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Mining encompasses various algorithms such as clustering, classification, association rule mining and sequence detection.

Traditionally, all these algorithms have been developed within a centralized model, with all data being gathered into a central site, and algorithms being run against that data. Privacy concerns can prevent this approach - there may not be a central site with authority to see all the data. We present a privacy preserving algorithm to mine association rules from several sites (IDSs) that partitioned horizontally which all sites have the same features, but each site has information on different entities. The goal is to produce association rules that hold globally, while limiting the information shared about each site (IDS).

2. LITERATURE REVIEW

2.1. Data mining

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of systematic software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

- Classes: Stored data is used to discover data in determined groups
- Clusters: Data features are grouped according to reasonable relationships or consumer favorites.
- Associations: Data can be mined to detect associations.
- Sequential patterns: Data is mined to anticipate behavior patterns and trends.

More thorough studies of data mining can be found in [1].

2.2. Association Rules

Let F = {F_1, F_2,…, F_m} be a set of features. Let D, be a set of database transactions where each transaction T is a set of features such that T $\subseteq$ F. Each transaction is associated with an identifier, called TID. Let A be a set of features. A transaction T is said to contain A if and only if A$\subseteq$ T. An association rule is an association of the form A → B, where A $\subset$ F, B $\subset$ F , and A $\cap$ B = φ. The rule A →B holds in the transaction set D with support s, where s is the percentage of transactions in D that contain A $\cup$B. This is taken to be the probability, P (A | B). The rule A → B has confidence c in the transaction set D, where c is the percentage of transactions in D containing A that also have B. This is engaged to be the conditional probability, P (B|A). That is,

Support (A→B) = P (A$\cup$B)

Confidence (A→B) = P (B|A) = $\frac{P(\mathrm{A,B})}{P(\mathrm{A})}$

Rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf) are called robust. A set of features is referred to as a feature set. A feature set that contains k features is a k-feature set. The happening frequency of a feature set is the number of transactions that contain the feature set. If the relative support of a feature set F satisfies a defined minimum support threshold, then F is a frequent feature set. The set of frequent k-feature sets is commonly denoted by L_k . From Equation above, we have

Confidence (A→B) =

$$\frac{support(A\cup B)}{support(A)} = \frac{support\_count(A\cup B)}{support\ count(A)} \quad (1)$$

Equation shows that the confidence of rule A→B can be easily derived from the support counts of A and A$\cup$B. That is, once the support counts of A, B, and A$\cup$B are found, it is straightforward to derive the corresponding association rules A→B and B→A and check whether they are robust. Thus the problem of mining association rules can be reduced to that of mining frequent feature sets.

In general, association rule mining can be viewed as a two-step process [2]:

1. Find all frequent feature sets: By definition, each of these feature sets will happen at least as frequently as a defined minimum support count.
2. Create robust association rules from the frequent feature sets: By definition, these rules must satisfy minimum support and minimum confidence.

2.3. Apriori Algorithm

As shown in [3, 4], Apriori Algorithm, can be used to create all frequent feature set:

Pass1:

1. Create the candidate feature sets in C_1.
2. Save the frequent feature sets in L_1.

Pass k:

1. Generate the candidate feature sets in C_k from the frequent feature sets in L_{k-1}.
2. Scan the transaction database to determine the support for each candidate feature set in C_k.
3. Save the frequent feature sets in L_k.

The pseudo code of Apriori algorithm is as follow:

```
Variable $C_k$: candidate feature sets of size k
Variable $L_k$: frequent feature sets of size k
L1 = {frequent items}
//prune candidates: For (k = 1; $L_k$ !=∅; k++) do begin
    // JOIN STEP: join $L_k$ with itself to produce $C_{k+1}$
    // PRUNE STEP: discard (k+1)- feature sets from $C_{k+1}$
    that contain non frequent k-feature sets as subsets
$C_{k+1}$ = candidates generated from $L_k$ candidates
//prove candidates: For each transaction t in database do
begin
    Increment the count of all candidates in $C_{k+1}$ that are
    contained in t
    $L_{k+1}$ = candidates in $C_{k+1}$ with min_sup
end
return ∪k Lk
```

2.3. Secure Sum Protocol

Secure Sum securely calculates the sum of values from indivisual sites. Assume that each site i has some value v_i and all sites want to securely compute $v = \sum_{l=1}^{m} v_l$ where v is known to be in the range [0…n]. Secure Multiparty Computation (SMC) could be used to calculate secure sum as follows [5]:

1. Site 1 creates a random value (r) and calculate sum of his value and random value as s_1.
2. Site 1 sends s_1 to Site 2.
3. Each site i take from previous site s_{i-1} and send to next site the sum of his value and s_{i-1} as s_i.
4. Site m sends s_m to site 1.
5. Site 1 calculate $v = s_m - r$;

3. PROPOSED DESIGN

In this approach, we consider KDD 99 as our organizations datasets and try to collaborative mine association rules from their datasets and use these rules to distinguish attacks from normal behaviors generally[6]. For instance we mine association rules to distiguish Neptune attack form other attacks and normal behaviors.

The steps of mining association rules are as follow:

- We try to select relevance features for attacks such as Neptune attack. We determine the contribution of 41 features in KDD 99 intrusion detection datasets to attack detection (or discremination of normal behavior from attacks). To this end, we use an approach based on information gain. Based on the entropy of a feature, information gain measures the relevance of a given feature. If the feature is relevant, in other words highly useful for an accurate determination, calculated entropies will be close to 0 and the information gain will be close to 1. Since information gain is calculated for discrete features, continuous features are discretized with the emphasis of providing sufficient discrete values for detection. List of features for which Neptune attack is selected most relevant is as follow[7]:
  Features # : 4,25,26,29,30,33,34,35,38,39.

These relevant features in KDD 99 are flag (Discrete), serror_rate (continuous), srv_serror_rate (continuous), same_srv_rate (continuous), diff_srv_rate (continuous), dst_host_srv_count (continuous), dst_host_same_srv_rate (continuous), dst_host_diff_srv_rate (continuous), dst_host_serror_rate (continuous), dst_host_srv_serror_rate (continuous);

- We consider features happenings and weights instead of just their happenings in the dataset to calculate their support and confidence. Thus, our suggested fraework reflects not only number of records supporting the feature sets, but also their degree of significance in the dataset.

We calculate weight of features values, using their frequencies, because values of features have more happenings, have more weights for support and confidence computation.

- After weighting values of features, we normalize them using z-score normalization method[8].
- We calculate weighted support and confidence as below:

Let a dataset D consists of a set of transactions T= {t_1 t_2… t_n} with a set of features F= {f_1, f_2… f_n}. A fuzzy dataset D′ contains fuzzy transactions T'= {t'_1 ,t'_2 ,...,t'_n } with fuzzy sets associated with each features in F, which is recognized by a set of semantic labels L={l_1 ,l_2 , ...,l_n }. We give a weight w to each l in L associated with f. Each label l_k for feature f_j would have associated with it a weight, i.e. a pair ([f[l]], w) is called a weighted feature where $[f[l]]\in L$ is a label associated with feature f and $w\in W$ is the weight associated with label l. More thorough studies of these concepts definitions can be found in [9].

Definition 1: Fuzzy Feature Weight is a value allocated to each fuzzy set. Weight of a fuzzy set for a feature f_j is denoted as $f_j[l_k[w]]$.

Definition 2: Fuzzy Feature set Transaction Weight is the product of weights of all the fuzzy sets associated to features in the feature set which existing in a single transaction. Fuzzy feature set transaction weight for a feature set (X, A) can be calculated as:

$$X = \prod_{k=1}^{|L|} t'_i[f_j[l_k[w]]] \qquad (2)$$

Definition 3: Fuzzy Weighted Support (FWS) is the sum of Fuzzy Feature set Transaction Weight of all the transactions in which feature set is present, divided by the total number of transactions. It is calculated as:

$$\mathrm{FWS}(X) = \frac{\sum_{i=1}^{n}\prod_{k=1}^{|L|} t'_i[f_j[l_k[w]]]}{n} \qquad (3)$$

Definition 4: Fuzzy Weighted Confidence (FWC) is the proportion of sum of chooses satisfying both $X \cup Y$ to the sum of chooses satisfying X with $Z = X \cup Y$. It is calculated as:

$$FWC(X \rightarrow Y) = \frac{FWS(Z)}{FWS(X)} == \sum_{i=1}^{n} \frac{\prod_{k=1}^{|Z|} {}_{(\forall[z[w]]\in Z)} t'_i[z_k[w]]}{\prod_{k=1}^{|X|} {}_{(\forall[i[w]]\in X)} t'_i[x_k[w]]} \qquad (4)$$

- Up to this point, sites mine their local association rules and wish to collaborate together to mine global association rules. Therefore, each site creates an array of features to store support of them. Each local frequent feature has non-zero value in the array and non-frequent feature has zero value in the array. Protcol performed between them is as Secure Array Sum Protocol.

Secure Array Sum Protocol:

1. Site 1 creates two array of local supports (x_1) and random values (r). Then, calculate sum of two arrays as s_1.
2. Site 1, sends s_1 to Site 2.
3. Each site i take from previous site s_{i-1} and send to next site, the sum of his array (x_i) and s_{i-1} as s_i.
4. Site m sends s_m to site 1.
5. Site 1 calculate $v = s_m – r$;
6.

P1: creates x_1 and r_1 array(x_1 is array of local supports and r_1 is array of random values);

$S1 = x_1 + r_1$;

1) $P_1 \rightarrow P_2 : S_1$

2) P_2: creates x_2 array.

$S_2 = x_2 + S_1$;

$P_2 \rightarrow P_3$: S_2;

3) P_3: creates x_3.

$S_3 = x_3 + S_2$;

$P_3 \rightarrow P_1$: S_3

P_1: calculates $V = S_3 – r$;

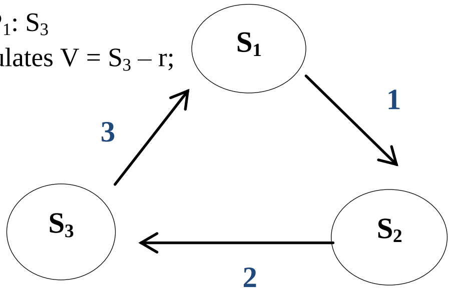

Figure 1. Secure Array Sum Protocol

- Ultimately, Site 1 can mine assocation rules from result.

4. RESULTS AND DISCUSSIONS

The rules generated using the proposed approach and shown below. After minnig association rules, we randomely select parts of 10% of kddcup.test which includes 521 attack and 330 normal records as test records.

Then, we test association rules and calculate F-value of results on the records as follows[10]:

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (5) \qquad \text{Recall} = \frac{TP}{TP+FN} \qquad (6)$$

$$F-value = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \qquad (7)$$

Where these values are $\in \{0, 1\}$. After calculating F-value, we use 3-step algorithm to limit association rules:

1. Sort association rules based on F-values and select association rules that have F-values more than 0.95.
1. Filter rules that have the most false negative (FN) ratio.
2. Sort rules based on False Positive (FP) Ratio and Filter rules that have FP Ratio more than 1%.

After this, we have 10 rules for neptune attack as follows:

39 → 33 (Sup = 0.025 , Conf =0.91)
39 → 4 (Sup = 0.02, Conf = 0.85)
39 → 25 , 30 (Sup = 0.05, Conf = 0.75)
29 → 4 (Sup = 0.041,Conf = 0.87)
39 → 4 , 30 (Sup = 0.034, Conf = 0.72)
39 → 4 , 33 (Sup = 0.036, Conf = 0.64)
39 → 4, 9 (Sup = 0.042, Conf = 0.86)
39 → 25 , 33 (Sup = 0.053,Conf = 0.76)
39 → 25 , 7 (Sup = 0.05 , Conf = 0.7)
39 → 33 , 9 (Sup = 0.048,Conf = 0.89)

Association rules such as 39 → 33 indicates, if features 39 then 33 occur, Neptune attack heppens. Namely, if feature 39 is 1 and feature 33 is 1 or feature 39 is 0 and feature 33 is 0/1, Neptune attck occurs (1→ 1 , 0 → 1 , 0 → 0).

As shown above, features act as binary features. But, for testing, we use fuzzy behaviors of features. Thus, we calculate the average of weights of features values as thresholds. Then,

- $feature\ value \geq threshold$ 1
- $feature\ value < threshold$ 0

Ultimately, by selecting these rules we have the most coverage of attacks for detection and FP ratio = 0.036, FN ratio = 0.0038 and detection rate = 0.996.

5. CONCLUSION

In this paper a generalized approach for mining weighted association rules from KDD intrusion detection dataset with binary and fuzzy features has been proposed. For instance, a number of association rules have been derived for Neptune attack. The approach used to mine association rules in collaborative IDSs by preserving privacy .It is effective to analyze the database containing discrete and continuous features with weighted settings. Here the poor rules having less support and confidence value have also been removed. The association rules generated will guide the IDS in evolving better rules to identify Neptune attacks.

Acknowledgment

I would like to thank Mr. Erfan Khosravian and Ms. Mahsa Saeidi for valuable feedbacks and Comments.

REFERENCES

[1]. UCL ANDERSON: School of management, *http://www.anderson.ucla.edu/ faculty /jason. frand/teacher/ technologies /palace/ datamining .htm*

[2]. Jiawei Han , Micheline Kamber: *Data Mining Concepts and techniques- Second Edition*: University of Illinois at Urbana- Champaign.

[3]. Bodon, F.: *A Fast Apriori implementation*. In: ICDM Workshop on Frequent Feature set Mining Implementations, vol. 90, Melbourne, Florida, USA (2003)

[4]. University of Regina: Department of Computer Science, *http://www2.cs.uregina.ca/~dbd/cs831/ notes / feature sets/featureset_apriori.html*

[5]. Charu C.Aggrawal, IB T.J. Watson Research center, USA; Philip S.Yu, University of Illinois at Chicago, USA: *Privacy Preserving Data Mining Models and Algorithms*;

[6]. S. Hettich, S.D. Bay, the UCI KDD Archive. Irvine, CA: University of California, Department of Information and Computer Science, *http://kdd. ics.uci.edu, 1999.*

[7]. H. Güneş Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood: *Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets*: Dalhousie University, Faculty of Computer Science, 6050 University Avenue, Halifax, Nova Scotia. B3H 1W5

[8]. Murat Kantarcioglu, Chris Clifton: *Privacy Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data*: Purdue University.

[9]. Tao, F., Murtagh, F., Farid, M.: *Weighted Association Rule Mining Using Weighted Support and Significance Framework.* In: Proceedings of 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 661–666, Washington DC (2003).

[10]. Lambert Schaelicke, Thomas Slabach, Branden Moore, and Curt Freeland, *Characterizing the performance of network intrusion detection sensors*, Proceeding of Recent Advances in Intrusion Detection, 6th International Symposium, (RAID 2003) (Pittsburgh, PA, USA) (G.Vigna, E.jonsson, and C.Kruegel, eds.), Lecture Notes in Computer Science, Springer- Verlag Heidelberg, September 2003, pp. 155-172.