 doi:10.20944/preprints201909.0040.v1

Application of data mining on web usage data for security: WebSecuDMiner

*Muhammad Zia Aftab Khan[1], Jihyun Park[2]
1. *Assistant Professor, Graduate Scool of Legal Affairs & Business Administration*
2. *Professor, Graduate Scool of Legal Affairs & Business Administration*
Youngsan University Busan South Korea
Corresponding author: ziaaftab3@gmail.com

Abstract

Purpose: The purpose of this paper is to develop WebSecuDMiner algorithm to discover unusual web access patterns based on analysing the potential rules hidden in web server log and user navigation history.
Design/methodology/approach: WebSecuDMiner uses equivalence class transformation (ECLAT) algorithm to extract user access patterns from the web log data, which will be used to identify the user access behaviours pattern and detect unusual one. Data extracted from the web serve log and user browsing behaviour is exploited to retrieve the web access pattern that is produced by the same user.
Findings: WebSecuDMiner is used to detect whether any unauthorized access have been posed and take appropriate decisions regarding the review of the original rights of suspicious user.
Research limitations/implications: The present work uses the database which is extracted from web serve log file and user browsing behaviour. Although the page is viewed by the user, the visit is not recorded in the server log file, since it can be access from the browser's cache.

Keywords: Data mining, Security, Association rule, ECLAT.
Paper type: Research paper

Introduction

In today's business environment almost all companies have their computers connected to the public Internet. As the number of companies with computers and services accessible to the Internet increases, a corresponding increase in the number of attacks against these businesses is also observed. There are external intruders, who are unauthorized users of the machines they attack, and internal intruders, who have permission to access the system with a number of restrictions (Elsheikh, 2008; Kowalski and Beheshti, 2006). Recently, the problem of information security on the Web has becoming an important research issue. Web data mining techniques can be applied to discover and extract information from Web documents and services.

An anomaly or malicious behavior of user is during web browsing is a main cause of internet crime. Web usage mining technique is useful in predicting and investigates the user behavioral from web log files (Singh and Jain, 2014).
When a user navigates through the web, his/her clicks are recorded in web log file. Analyzing these log files using data mining reveal many interesting patterns (Mahoui,

 doi:10.20944/preprints201909.0040.v1

et al., 2001). The purpose of web mining is to identify potential users for the web (Joshila et al., 2011; Thakkar and Rao, 2013). Web usage mining technique is useful in predicting and investigates the user behavioral from web log files (Singh and Jain, 2014).

One of the basic methods of web mining is association rules that indicate relationships among common use of web pages (Daş and Türkoğlu, 2009). The association mining task consists of identifying the frequent itemsets, and then forming conditional implication rules among them (Zaki et al., 1997). Many algorithms were proposed to find frequent patterns in transaction databases namely Apriori (Kumar and Rukmani, 2010), FP-growth (Gupta, P. and Mishra, 2011) and ECLAT (Zaki and Gouda, 2003). ECLAT was proposed by Zaki in 1997 and he proved that ECLAT is better than Apriori Algorithm. ECLAT is a classical algorithm for mining frequent itemsets, which is based on vertically out databases (Zaki et al, 1997). It is greatly different from those algorithms based on horizontally out databases, such as algorithm Apriori and FP-Growth. ECLAT uses vertical data format for frequent pattern mining. It is depth first search technique. It is proved that ECLAT is better than Aprioria Algorithm (Choubey et al., 2011). It needs less database scan compare to apriori, ECLAT is faster than apriori (Solanki and Soni, 2015).

This paper focuses on using association rule algorithms. The algorithms based on Equivalence Class Transformation (ECLAT) applied on association rule mining to detect unusual access behavior pattern and discover suspect user behavior. The specific objective is to develop WebSecuDMiner system algorithm for mine the Web usage data to discover unusual access behavior pattern, then take appropriate decisions regarding the review of the original rights of suspicious users and inform the security system administers.

The following two examples illustrate the idea:

Example1: A research agency includes comprehensive systems for information capturing, sharing and managing different projects. Each project has a specific research group. Each group has authorization to update or modify the research information in only while for bidding for other groups to do so and vice- versa and make sits technical reports available to a certain group of users such as students, researchers, etc. A smart use of the results included in the technical reports by concurrent agencies has jeopardized the existence of the future work of the original research agency.
Example2: New hired employee to a research center try to gets online access to local direction that contain sensitive data. He uses his privileges to spy on the center trying to extract maximum information about the new patents the center is working on. This information will be delivered to competitor companies. In these examples, the data/service is manipulated by legitimate users. It is a misuse that will trigger the review of the original access rights attributed initially to the user. Online mining where activities of new users are monitored online and reaction to misuse of privileges is generated automatically and immediately

This paper is organized as follows: Section 2 describes Related work background objects related to web usage mining and the association rules method that is used in this study. The analysis and design of methodology is described in Section 3. WebSecuDMiner Algorithm is presented in Section 4. Finally, the conclusion is represented in the section 5.

 doi:10.20944/preprints201909.0040.v1

Related Work

Web Usage Mining

The Web Usage Mining is the application of data mining technique to discover the useful patterns from web usage data. It can discover the user access patterns by mining log files and associated data of particular web site. Web usage mining refers to the automatic discovery and associated data collected or generated as a result of user interactions with Web resources on one or more Websites. The goal is to capture, model, and analyze the behavioral patterns and the profiles of users interacting with Web sites. The discovered patterns are usually represented as collections of pages, objects, or resources that are frequently accessed by groups of users with common needs or interests (Daş, and Türkoğlu, 2009). Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data. The usage data captures the identity or origin of Web users along with their browsing behavior at a Web site.

In web usage mining, this dataset is the huge web data. Web data contains different kinds of information, including, web structure data, web log data, and user profiles data. Web mining is the application of data mining techniques to extract knowledge from web data, where at least one of structure or usage data is used in the mining process (Daş and Türkoğlu, 2009). Web Usage Mining consists of three phases which are data processing, pattern discovery and pattern analysis. The data processing phase has two parts called data cleaning and filtering. Filtering is the most important task in web usage mining since the quality of mined patterns depends on this directly. In the pattern discovery phase, Special pattern discovery algorithms applied on raw data which is output of the data processing phase. In the pattern analysis phase interesting knowledge is extracted from frequent (Dimitrijević et al., 2010).

Association Rule mining

Association rule mining discovery techniques are generally applied to databases of transactions where each transaction consists of a set of items. In such a framework the problem is to discover all associations and correlations among data items where the presence of one set of items in a transaction implies (with a certain degree of confidence) the presence of other items. In context of web usage mining (Mishra and Choubey, 2012).
Association Rules are probably the most elementary data mining technique and at the same time, the most used technique in Web Usage Mining. When applied to Web Usage Mining, association rules are used to find associations among web pages that frequently appear together in users' sessions. The terms used in these rule are:

- *Support:* The support of an association rule X implies that Y is the percentage of transaction in the database that consists of X U Y.

- Confidence: The confidence for an association rule X implies that Y is the ratio of the number of transaction that contains X U Y to the number of transaction that contains X (Mishra and Choubey, 2012).

2.2.1 Association Rules Algorithm: Definition

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 September 2019 doi:10.20944/preprints201909.0040.v1

The problem is stated as follows, Let $I = \{i_1, i_2 \ldots i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. A unique identifier TID is given to each transaction. A transaction T is said to contain X, a set of items in I, if $X \subseteq T$. An association rule is an implication of the form "$X \Rightarrow Y$", where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ has a support s in the transaction set D is s% of the transactions in D contain $X \cup Y$. It is said that the rule $X \Rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. In other words, the confidence of the rule means that the consequent Y is true under the condition of the antecedent X (Kotsiantis and Kanellopoulos, 2006). Confidence and support are significant measures of rule interestingness and they reflect usefulness and certainty of rule respectively (Kaur and Grag, 2014). (Dimitrijevic and Bosnjak, 2010).

The selection of the association rules depend upon these two values because the selected rules should have support and confidence greater than the respective threshold values. This is done through two-step approach (Tan et al., 2010):

1. Frequent Itemset Generation
   - Generate all itemsets whose support ≥ minsup
2. Rule Generation
   - Generate high confidence rules from each frequent itemset

ECLAT Algorithm

The equivalence CLASS Transformation (ECLAT) Algorithm is one of associations rule mining Algorithm (Zaki and Gouda, 2003). The algorithm finds frequent patterns by a depth-first search. It uses a vertical database representation and counts the itemset supports using the intersection of Tids. However, because of the depth-first search, pruning used in the Apriori algorithm is not applicable during the candidate itemsets generation (Kaur and Grag, 2014; Solanki and Soni, 2015).

In the ECLAT (Equivalence CLASS Transformation), mining frequent patterns from a set of transactions in item-TID-set format (that is, {items: TID-set}), where item is an item name, and TID-set is the set of transaction identifiers containing the item. This format is known as vertical data format. (Zaki and Gouda, 2003).

In the ECLAT algorithm the mining steps are as follow:

- First, transform the horizontally formatted data to the vertical format by scanning the data set once. The support count of an item set is simply the length of the Tids of the item set. Starting with k = 1, the frequent k-item sets can be used to construct the candidate (k+1)-item sets based on the Apriori property.
- Second, the computation is done by intersection of the Tids of the frequent k-itemsets to compute the Tids of the corresponding (k+1)-itemsets.
- Third, this process repeats, with k incremented by 1 each time, until no frequent item sets or no candidate itemsets can be found. Besides taking advantage of the Apriori property in the generation of candidate (k+1)-itemset from frequent k-itemsets, another merit of this method is that there is no need to scan the database to find the support of (k+1) itemsets.

Preprocessing

The users' access to Web sites are stored in web server log files. But the data stored in these log files do not present an accurate picture of the users' access to the Web site. So the preprocessing of web log data is a pre-requisite phase before it can be used for mining tasks (Pani, 2011; Payal and Nimavat, 2015). The most important task of the web log mining process is data preparation. A Web server usually registers a Web log entry for every access of a Web page. Due to large amount of irrelevant information in the Web log, the original log can't be directly used in the Web log mining procedure. Purpose of data preprocessing is to offer structural, reliable and integrated data sources to pattern discovery. The preprocessing module converts the web server log file, which normally is in ASCII forma or plain text format, into a database like format (Joshila et al., 2011).

- Web server log

A web server log file is a simple plain text file which records information each time a user requests a resource from a web site. This file is opened when the web services of a server starts and remain open as the server responds to user requests (Joshila et al., 2011). The entries of a web log file consists of several fields which represent the date and the time of the request, the IP number of the visitor's computer (client), the URL request , the HTTP status code returned to the client, and so on. The log data collected at Web access or application servers reflect navigational behavior knowledge of users in terms of access patterns.

In this research paper four server log file entries are selected, include:

(1)User Identification: User password.

(2) Date (Date of last visit): the time duration from when the user accessed last, e.g. today, yesterday, the day before yesterday, etc.

(3) URL: URL of the page visited, the Web server log contains a complete history of file access by clients. Due to local caches not all page requests made to a server are recorded in the server log file. Since the browser finds in its cache a copy of a document being requested by the user, the request is not made to the server and the stored copy of the document is displayed. Therefore, although the page is viewed by the user, the request is not recorded in the server log file, since it can be found in the browser's cache. If the user disables the browser's cache, then the user's requests can be recorded

(4) Status: Status code is status field that is set by the web server and indicates the action taken in response to a request. For example (Codes from 200 through 299 indicate success, 300 through 399 indicate some form of redirection, 400 through 499 indicate an error serving the particular request and 500 through 599 indicate a problem in the web server).

System Analysis and Design

System Analysis

Web server log are files which stores user click streams whereas navigating an internet website. A number of these knowledge area unit supernumerary for the analysis method and will have an effect on the detection of net attacks. Therefore, preprocessing is necessary, because Log file contain noisy & ambiguous data which may affect results of mining process. Some of web log file data are unnecessary for analysis process and could affect detection of web attack. The purpose of data preprocessing is to improve data quality and increase

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 September 2019 doi:10.20944/preprints201909.0040.v1

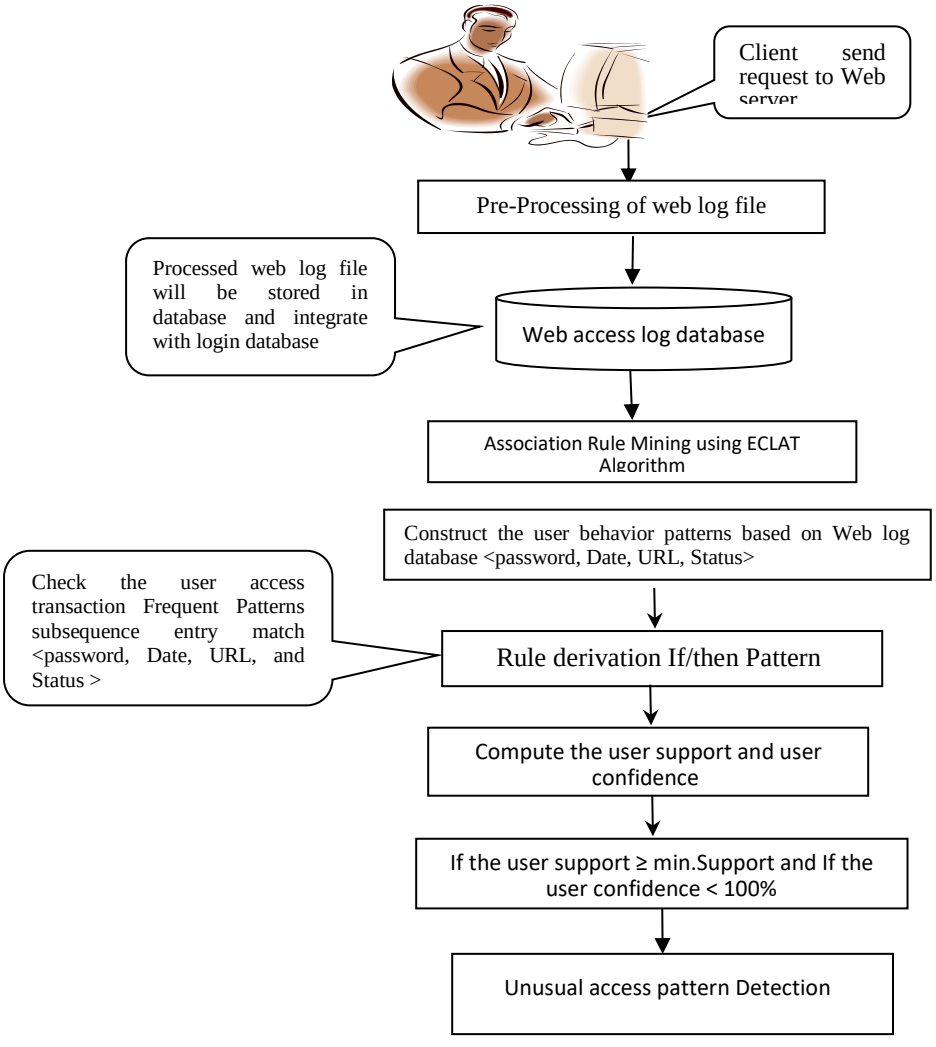
mining accuracy. Preprocessing consists of field extraction, data cleansing, user identification, and session identification.

System Architecture

Figure 1 show WebSecuDMiner System Architecture. The system Architecture represents the conceptual model of the system, which defines the structure, behavior, and more views of a system, organized in a way that supports reasoning about the structures of the system.

System Description

The proposed system algorithm is developed to discover unusual access patterns. This system searches the interesting relationships and web access patterns from the web log file based on ECLAT Algorithm. Firstly, this system analyzes the Web log database. Second, it counts the support for each web access transaction entries attributes found. Then, it is compared with minimum support count. If it is less than minimum than support count is removed and others go on processing. And then, this system can again compare each of them with minimum support count and remove pairs which are less than minimum support count. IF-THEN association rule mining are used to generate web access patterns and behaviors based Web usage data in Web server log file. Web server log file captures the identity or origin of Web users along with their browsing behavior. Four server log file entry attributes was considers < password, Date, URL, Status>, when the presence of one set of Web access entries pattern in the transaction implies the presence of others with confidence 100 % for usual web access pattern, If the confidence < 100% then unusual web access patterns is occurred implies suspected behaviors of the user.



 doi:10.20944/preprints201909.0040.v1

Figure 1. Architecture of Proposed System

WebSecuDMiner Proposed System Algorithm

Steps involved in the proposed system algorithm include

Step1. Client sends the request to the server for the web pages he/she wants to access on the web.

Step 2. A Web server log is a file to which the Web server writes information each time a user requests a web site from that particular server. The server log file records all requests that are processed by the server. The data go through a preprocessing stage to clean the data from irrelevant or redundant data in Web server logs. Next, the output is web server log database file. This database integrated with the user login data to produce the Web access transaction database, which contains password, Date (Date of last visit), URL of the page(s) visited, and the Status (The status action is set by the Web server).

Due to local caches, not all page requests made to a server are recorded in the server log files. Therefore, although the page is viewed by the user, the request is not recorded in the server log file, because it is recorded in the browser's cache. In this proposed model, if the browser's cache is disabled; therefore, the user's requests can be recorded.

Four web server log access transactions entry attribute ware selected, are as follows:

(1) User Identification: User password.

(2) Date (Date of last visit): the time duration from when the user accessed last, e.g. today, yesterday, the day before yesterday, etc.

(3) URL: URL of the page visited. The Web server log contains a complete history of file access by clients. Due to local caches not all page requests made to a server are recorded in the server log file. If the user disables the browser's cache, then the user's requests can be recorded.

(4) Status: Status code is the status field, which is set by the web server and indicates the action taken in response to a request. For example (Codes from 200 through 299 indicate success, 300 through 599 indicate unsuccessful.

- Preprocessed server Log [Definitions]

1. Definition 1. Let U be a set of attributes that identify the Web server log access transaction entry, $U \subseteq L$, such that U= {password, URL of page visit, Date of last visit, Status action}. The value for each attribute in a class set U order set, where The class set U order forms a Web server log access transaction database < password, Date, URL, Status>.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 September 2019 doi:10.20944/preprints201909.0040.v1

2. Definition 2. Let L= {l_1, l_2 ……., l_m} be the set identified a Web server log access transactions entry attributes ,where l Є L is define as a tuple, l_i = {password$_i$, Date$_i$ URLl_i, Status$_i$}, where the transactions entry l_i Є L , such that for password $_i$ = user password , Date$_i$ = date-of-last visit, URL$_i$= URL of the page visited, and Status$_i$=status code given by the server, for $l_i \subseteq$ L, where 1< i<m. Identification is based on the class set identifier (password).

Step3. Apply the ECLAT Algorithm. This algorithm mines the filtered database and it looks for frequent patterns

- ECLAT ALGORITHM: Terminology

(i) Fk is defined as Web access database having F_k = {I_1, I_2, .., I_m}
(ii) Φ denotes empty database, where Web server log access
iii) U denotes set of web server access transactions entries.
ECLAT Algorithm in pseudo code
Input: F_k = {I1.. In} frequent k web access transaction entry attributes
Output: F$_{|R|}$ Frequent Item Sets
Bottom-Up (F_k);
for all I_i ϵ F_k do
F_{k+1} = ϕ
For all I_j ϵ F_k, i < j do
N = I_i ∩ I_j ; // I_i and I_j Both should be from same equivalence class
if N.sup ≥ minsup then
F_{k+1} = F_{k+1} U {N}; F$_{|R|}$ = F$_{|R|}$ U {N}
if F_{k+1}!= ϕ ; then
Bottom-Up(F_{k+1});
end;

In this algorithm, F_k stores the number of user access transaction entry data as input. Output contains the web access transaction entry patterns which frequently occurred. First F_{k+1} starts with the prefix {} and the search tree is actually the initial search tree. To divide the initial search tree, it picks the prefix {Password}, generate the corresponding equivalence class and does frequent of web access transaction entries mining in the sub tree of all of web access transaction entry patterns containing {Date}, in this sub tree it divides further into two sub trees by picking the prefix{Password, Date }. The first sub tree consists of all web access transaction entry patterns containing the prefix {Password, Date}, and this process is recursive until all web access transaction entries pattern in the initial search tree are visited, the search tree {Password, Date, URL, Status}. Now compare the support of web access transaction entries pattern with the minimum support. Put all those web access transaction entries in F_{k+1} . F_{k+1} contains web access transaction entries pattern. Again check that F_{k+1} is empty or not. If it is not empty then bottom up approach will apply on F_{k+1}.

The search strategy employed by an algorithm shape how the lattice structure is traversed during the frequent web access transaction entry attributes generation process. In Equivalence Classes the traversal first partition the lattice into disjoints groups of equivalence classes). A frequent web access transaction entries generation algorithm searches for frequent web access entries within a particular equivalence class first before moving to another equivalence class. Equivalence classes defined according to the prefix. In this case, two entries belong to the same equivalence class if they share a common prefix.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 September 2019 doi:10.20944/preprints201909.0040.v1

Example1 (using the formal definition 2): Let X, Y, Z, M $\subseteq$ U where X = user password, Y=URL, Z=Date of visit, and M=Status (examples of Users Transaction Entries). In the prefix-based approach, the algorithm can search for frequent web access transaction entries. Starting by prefix X before looking for those starting with prefixes Y, Z, M, and so on. Both prefix-based equivalence classes can be demonstrated using the search tree structure shown in Figure (2).

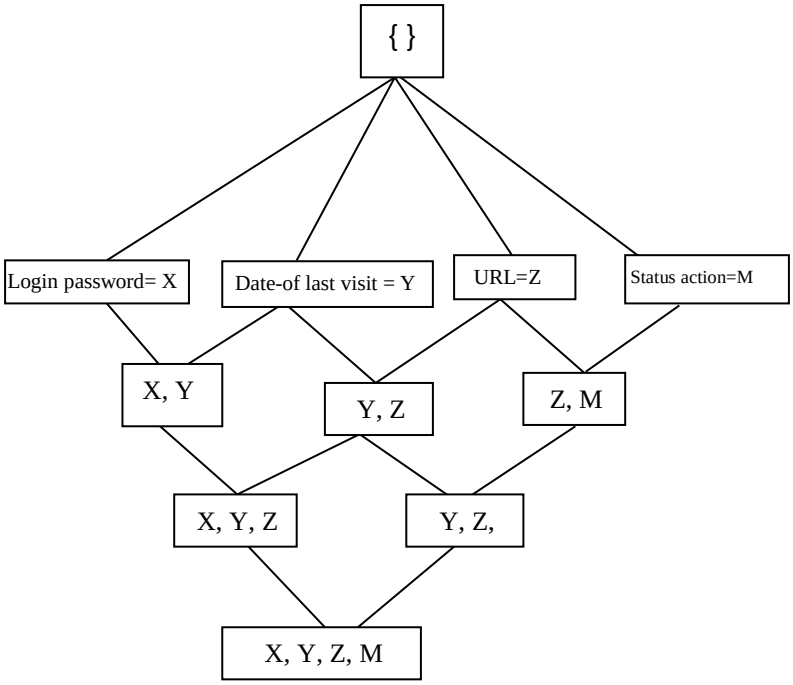


Figure 2. The search tree structure in WebSecuDMiner

Step 4. IF-THEN association rule generation well is used in this study. The application of If/then association rule mining is to discover the associations and correlations among web user access transaction entry attributes { login password, date of last visit, URL (page visited), and Status (status code sent by the server}, where the presence of one set of web access entries attribute in the transaction implies the presence of others with 100% confidence, and minimum support ≥ 2.

Example1 (using the formal definition association rule): Let X, Y, Z, M $\subseteq$ U where X = user password, Y=Date of visit, Z = URL, and M = Status. The association rule evaluation metrics

– Support (S): Fraction of transactions that contain both X and Y etc.
– Confidence (C): Measures how often items in Y appear in transactions that contain X, etc.

$$C(X \rightarrow Y) = \frac{S(X \cup Y)}{S(X)}$$

The, association rules show relationship among Web access transaction entries. < password, Date, URL, Status > when the presence of one set of Web access entries pattern in the transaction implies the presence of others with minimum support >= 2, and confidence < 100% then unusual user access patterns is occurred implies suspected behaviors of the user or the user has no authority to access the Web (see figure 3).

 doi:10.20944/preprints201909.0040.v1

Start.
1. *check the rule (If (login password= password,) and (Date = Date-last-visit) and (URL=page visited) Then (Status =200)*
2. *compute the mini* support (S) and confidence (C) *of the rule*
   - 2.1. *if the confidence (C) the rule < 100%;*
   - 2.2. *If yes, then unusual pattern* then
     Ask the user to login again then goto 1. // the user allow to login two times. After the third trial reject the request and notify the administrator about it
   - 2.3. *Stop.*
3. *If no, Processing the user request*

Figure 3. IF-THEN rule format in WebSecuDMine

5. Conclusion

In this study, we have presented a new mining algorithm called WebSecuDMiner for security. The mining algorithms monitor user's access to web servers in order to identify anomaly or suspected user behavior, using The ECLAT Algorithms. Log files are the best source to know the user behavior, because when a user navigates at any web site and every click is recorded in web log file. Our research in future is implementation of WebSecuDMiner algorithm in such a way that produces precise and accurate patterns for suspected user identification and unauthorized access.

References

Choubey, A., Patel R., and Rana, J. L. (2011)," A Survey of Efficient Algorithms and New Approach for Fast Discovery of Frequent Itemset for Association Rule Mining (DFIARM)", *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 1 No.2, pp. 62-67.

Daş, R., & Türkoğlu, İ. (2009). Extraction of interesting patterns through association rule mining for improvement of website usability. *Istanbul University, Journal of Electrical & Electronics Engineering (IU-JEEE)*, Vol. 9 No. 2, pp. 1037-1046

Dimitrijevic, M.and Bosnjak, Z. (2010). " Discovering interesting association rules in the web log usage data". *Interdisciplinary Journal of Information, Knowledge, and ManagementI (JIKM), Vol. 5 No. 443, pp. 191-207.*

Elsheikh, S. (2008), " ACMW: Access Control Model on Web Environment", Proceedings of the World Congress on Engineering and Computer Science (WCECS 2008), October 22 - 24, 2008, San Francisco, USA.

Gupta, P. and Mishra, S. (2011), "Improved FP Tree algorithm with customized

web log preprocessing", International Journal of Computer science and technology (IJCST), Vol. 2 No. 3, pp. 30-32.

Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter*, Vol. 2 No. 1,pp. 58-64.

Joshila, L. K., Maheswari, V., and Nagamalai, D. (2011), "Analysis of web logs and web user in web mining", *International Journal of Network Security & Its Applications (IJNSA)*, Vol.3 No.1, pp. 99-110.

kaur , M. and Grag, U. (2014), "ECLAT Algorithm for Frequent Itemsets Generation", *International Journal of Computer Systems* , Vol. 1 No. 3, pp. 82-84. Available at http://www.ijcsonline.com/

Kumar, B. S and Rukmani , K.V. (2010)" Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms", *Int. J. of Advanced Networking and Applications*, Vol.:01 No. 06, pp. 400-404.

Payal Sagar, P. and Nimavat, A.V. (2015), "Web Usage Mining: Survey on Process and Methods", *International Multidisciplinary Research Journal*, Vol. 2, No. 5, pp. 1-4.

Kowalski, K. & Beheshti, M. (2006), "Analysis of Log Files Intersections for Security Enhancement", proceedings of the Third International conference in Information Technology: New Generations, IEEE, Las Vegas, NV, pp. 452 - 457.

Kotsiantis, S. and Kanellopoulos, D. (2006), " Association RulesMining: A Recent Overview, GESTS ", *InternationalTransactions on Computer Science and Engineering*, Vol.32 No. 1, pp. 71-82.

Mahoui, M., Bhargava, B., and Mohania, M. (2001), "Data Mining For Web Security: UserWatcher," Proceedings of the IC'2001 Conference, Las Vegas, USA.

Malviya, M. Jain, A., and Gupta, N. (2011), " Improving security by predicting anomaly user through web mining: a review*", International Journal of Advances in Engineering & Technology*, Vol. 1 No. 2, pp. 28-32.

Mishra, R., & Choubey, A. (2012)," Comparative Analysis of Apriori Algorithm and Frequent Pattern Algorithm for Frequent Pattern Mining in Web Log Data".

Pani, S. K. (2011), "Web Usage Mining: A Survey on Pattern Extraction from Web Logs", *International Journal of Instrumentation, Control & Automation* (IJICA), Vol. 1, No. 1, pp. 15-23.

Tan, P. N., Steinbach, M., & Kumar, V. (2005)," Association analysis: basic concepts and algorithms. Introduction to data mining", pp. 327-414.

Thakare, P. R. and Rao, K. H. (2013), "Anomaly Detection Based on Access Behavior and Document Rank Algorithm", *International Journal of Computer Trends and Technology (IJCTT)*, Vol. 6 No. 4, pp. 232-235.

Singh, A. P., & Jain, D. R. (2014), "A Survey on Different Phases of Web Usage Mining for Anomaly User Behavior Investigation" *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Vol. 3 No. 3,* pp. 70-75.

Solanki, S. and Soni, N. (2015), "Frequent Itemset Mining Using ECLAT with Relative Profit and Price", *International journal of computer techniques, Vol. 2 No. 3*, pp. 99-110.

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997), "New Algorithms for Fast Discovery of Association Rules". In KDD KDD-97), Vol. 97, pp. 283-286.

Zaki, M. and Gouda, K. (2003). "Fast Vertical Mining Using Diffsets", *Proc. Ninth ACM* SIGKDD '03, Int'l Conf. Knowledge Discovery and Data Mining (KDD '03), Washington, DC, USA, pp. 326-335.