CERIAS Tech Report 2001-20

Data Mining For Web Security: UserWatcher

Malika Mahoui[1], Bharat Bhargava[1], Mukesh Mohania[2]
Center for Education and Research in
Information Assurance and Security
&
[1]Department of Computer Science, Purdue University
West Lafayette, IN 47907
[2]Department of Computer Science, Western Michigan University

Data Mining For Web Security: UserWatcher *

M. Mahoui[1], B. Bhargava[1], M. Mohania[2]
[1]CERIAS and Department of Computer Science
Purdue University
West Lafayette, IN 47907
[2]Department of Computer Science, Western Michigan University
Kalamazoo, MI 49008

CERIAS TR 2001-20

Abstract **-** Data mining techniques have proved to be efficient for discovering interesting and useful patterns in large amount of data such as in Web documents. This paper investigates the use of mining techniques to secure Web access. We propose UserWatcher, a mining tool that integrates Web usage mining and Web content mining to find potential correlations between data that a user accesses and the data that he/she produces. Applied to security, UserWatcher can be used to detect information misusage by authorized users and to detect anomalies in current system policies.

Keywords: Security, Web Mining, Content Mining, Usage Mining, Information Misusage, Similarity Computation.

1 Introduction

With the exponential growth of the World Wide Web, Web data mining has emerged as a new research area to tackle the specific nature of different types of data used in the Web. This ranges from legal data to text documents and including hyperlinks and massive amounts of usage Web data [4, 11]. Security is an important issue in the process of controlling access to Web documents. Static Web access control approaches, where the number of users is predefined are no longer adequate to deal with the dynamic nature of the Web. Indeed, information available on the Web presents the great advantage of being accessible by an unlimited number of users. Dynamic Web access control approaches have been proposed to provide new users with access rights based on their credentials. This new approach includes relatively "simple" mechanisms used by Web servers to register new users. It also refers to new approaches proposed in studies such as in [9, 15, 19] to tackle the general problem of defining access policies to deal with request made by "strangers" to access sensitive data. The main idea is to provide mechanisms that allow a security system to check the validity of the credentials provided by the user such as using digital certificate made by a third party [9] and then infer the appropriate access rights based on the access policies defined by the system. We think that no matter how trustworthy are these credentials, a new user needs to pass through a probationary phase until the credentials provided have been enforced by the user behavior to the system. Another security risk that new users bring is the misuse of information or service. In fact we think that these two classes of risk exist for Web users but also for traditional operating systems. In later case, the number of users is relatively limited where in the former case this number is not bound; which increases the security risk. The following examples illustrate our idea:

Example 1: A research agency makes its 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.

Example 2: New hired employee to a medical research company gets access to local directories

* This work appears in the proc. of the international conference on Internet Computing IC'2001.
This research is supported by the Center for Education and Research in Information Assurance and Security and NSF grants EIA-9805693, CCR-001788, and CCR-9901712.

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.

Example 3: A Web server providing free Internet phone to users advertises for other non-free services. A new registered to the free service may misuse his/her rights by making very long calls every day, without using any of the non free services provided by the server.

In these examples, the data/service is manipulated by legitimate users. It is its misuse that will trigger the review of the original access rights attributed initially to the user. Mining user's behavior will allow to detect such situation and take appropriate decisions regarding the review of the original rights of suspicious users.

We identify two classes of mining user's behavior for security alert purposes. Online mining where activities of new users are monitored online and reaction to misuse of privileges is generated automatically and immediately. This type of mining is recommended for situations where an immediate reaction is necessary when the user misuses his/her rights. For example disconnecting a user that tries for a number of times during a session to access a very sensitive folder. On the other hand, offline mining of user behavior is recommended for situations where the system is not capable to detect whether a user behavior is suspicious or not. The system can only make recommendations and it is only the decision maker or system administrator who is capable to give a meaning to the mining results.

Applying data mining techniques to security has only been addressed recently [1, 3]. In this paper we are interested to use data mining as an alert tool for security system to help detect and filter out non desired users. We focus on the area of offline mining and in particular on proposing solutions for security issues similar to what we described in example 1. We propose a system called *UserWatche*r that uses both Web usage and Web content information, that we describe in section 2, to find out whether a user is exploiting the current server's data/services to publish similar data/services. Data extracted from server's user usage is exploited to retrieve similar data/services in the Web that is produced by the same user. This information is then provided to the decision maker to assess its relevance and determine whether the user rights must be altered or not. The challenging task in this process is the handling of Web documents: how do we determine what part of an accessed document is relevant to the user and how do we find user's documents that are similar. We choose to select a subset of keyphrases from the document to be its representative. This subset includes keyphrases automatically extracted from the documents. We use KEA an automatic keyphrase extraction system [7] to automatically extract keyphrases from the server documents accessed by the user, and also from the user documents that have been retrieved.

The rest of the paper is organized as follows. Section 2 briefly describes Web mining process and how our system uses Web mining for security alert. Section 3 presents an overall description of UserWatcher architecture. In section 4 and 5 we describe the different components of the system, namely data preparation and data analysis. In section 6 we conclude the paper and highlight some directions for future work.

2 Web mining and Web security

In recent years, Web mining has emerged as an important branch of data mining. This is mainly due to the tremendous amount of information available from the Web, which attracted many research communities, and the recent interest of e-commerce [11].

As described in the survey conducted in [11], Web mining has followed three main directions based on which part of the Web to mine. *Web content mining*, *Web structure mining* and *Web usage mining*. Web content mining describes the process of extracting useful information from millions of sources across the Web. This includes the content of Web documents as well as data/documents that are accessible from the Web.

Web content data can be a combination of unstructured data such as free texts, and semi-structured data such as text extracted from HTML files and structured data in the tables or database generated HTML pages. By large the information from the Web is non structured [4, 5, 8, 11].

Research conducted in Web content mining mainly led by IR and DB communities have focused on providing a higher level of organization to the data available on the Web.

Web structure mining uses the link structure of the Web for the purpose of organizing, visualizing and searching [2, 8].

Web usage mining exploits information derived from user interaction with the system to discover users patterns in order to understand and better serve the needs of Web users. Web usage data mainly includes information automatically generated by the system and stored in access logs. It also includes data collected from other sources resulting from the user interaction such as proxy server logs, registration data, user profiles.

Each of these Web mining categories concentrate on mining one main type of Web data but all three use the same subtasks to perform the mining process namely, *resource finding*, *preprocessing*, *pattern discovery* or *generalization* and *pattern analysis* [5, 11, 14]. Resource finding refers to the process of retrieving information that is accessible from the Web resources such as data in newsletters, or text data extracted from HTML documents. Data preprocessing refers to the set of transformations applied to the original information retrieved in the previous step to get it ready for the mining process. This includes cleaning information from irrelevant data such as stop words, properly identifying the items that are relevant to the mining process (see user identification in section 3); and presenting the information in a desired representation such as extracting textual information from HTML documents and transforming them into relational tables.

Using mining techniques for Web security needs to be investigated. Research conducted in [1, 3] focused on devising techniques for limiting the disclosure of mined information. We propose to integrate both Web usage mining and Web content mining in a security alert tool for discovering suspicious use of user's privileges.

3 System architecture

3.1 Approach

UserWatcher is an alert tool. Its main function is to detect if the part of the server's content that is accessible by the user has been used to publish products with similar content. Similar to the definition of Web content, server content refers to the broad range of data extracted from the server. This includes the text extracted from HTML documents, local documents (i.e. pdf, ps files) resulting from search queries or directly accessed from Web pages; and also data residing in local databases accessible from search queries.

We exploit the usage content to identify each user as well as the data that has been accessed. We distinguish three types of data accessible by the user:

- *Flat data:* which includes raw data related to search queries (search terms and result terms). Search results that are document references (hyperlinks) are not included in this category. Flat data also includes data gathered by the system which describes user preferences and helps in the identification of the user. This information is available from files such as user profile/preferences, registration data, etc.
- *Metadata*: includes metadata that features user access points in Web pages such as hyperlinks, Web page options (i.e. buttons) etc. This data is extracted from XML and HTML tags.
- *Document data*: which includes the content of all types of accessible documents from the server such as HTML/XML, ps and pdf documents. In the case of Web pages, also referred to as pageview [5, 14], document data includes all the information accessible from the page.

The task of retrieving user documents that include flat data is not difficult. The challenging task is to determine what part of the server document data is relevant to the user. Using document metadata accessed by the user is a partial solution to the problem. We also must include information that describes the content of the Web document. We use keyphrases automatically extracted from the document to be descriptors of the content of the document. Document metadata combined with flat data and automatically extracted keyphrases are used to search the Web for user documents that include the searched information. Once the user documents are retrieved, we are able to determine whether they are similar to the server documents or not. To compute the similarity between pairs of documents we use automatically extracted keyphrases combined with metadata extracted

from documents (if any) as the document features. High similarity scores are identified and corresponding user documents are recommended for the decision maker for further analysis.

metadata to describe the content of databases (extracted from the schema of the database). To summarize, server content is a combination of document data and metadata. User content on the

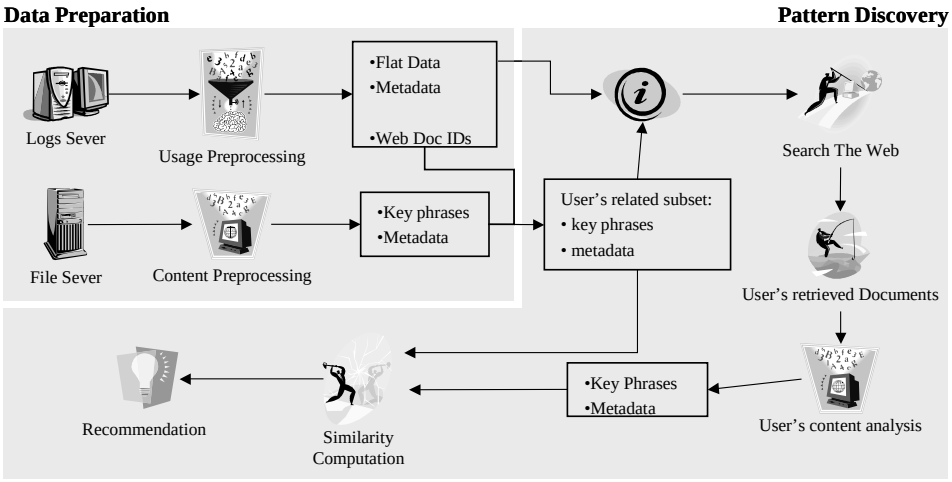

Figure 1: UserWatcher architecture

3.2 Architecture

The overall architecture of UserWatcher system is depicted in figure 1. Two main components are emphasized: *data preparation* and *pattern discovery*. Data preparation consists of the first two steps of a typical mining process namely data selection and data preprocessing. Structured data output from the processing of content data and usage data is fed to the pattern discovery component. In this part of the mining process, we distinguish three main components: retrieval of user documents, preprocessing of these documents and similarity computation. The result is output as a set of recommendations for the decision maker for further analysis.

4 Data preparation

4.1 Data selection

As described in section 2, we distinguish three main types of data: flat data, document metadata and document data. We assume that the content of a server is mainly composed of documents from which we can extract text. For non textual documents we assume that the server provide metadata describing the content. We also use other hand includes the three types of data. Note that from the user side, the document data refers to the portion of the server that the user has accessed; and for each accessed document, metadata refers to the document access points that the user has effectively clicked on.

4.2 Content preprocessing

Recall that the output of the server content preprocessing is used to retrieve user's documents and to compute documents similarity. The later involves the selection of a set of features for the vector representation of documents [17]. We use automatically extracted keyphrases from the documents combined with the documents metadata as the features of the document. We use Kea system [7] to automatically extract keyphrases from each document. Kea uses machine learning techniques to build a model of keyphrases in a collection and then applies this model to identify likely keyphrases within a document. The most frequent keyphrases are included in the document features set. The number of keyphrases extracted from each document is a variable parameter; setting it to five has shown good results [10, 16].

The advantage of considering most frequent keyphrases instead of a subset of document words or keywords has many advantages: keyphrases include one or more words and therefore they are more likely to capture the semantic of a document compared to using single words [12]. Moreover, the attributes used by Kea to automatically extract keyphrases have shown to be good measures in determining documents' keyphrases [10, 16]. These attributes are (1) the distance from the start of the document that the keyphrase first occurs and (2) the keyphrase's tf.idf (term frequency x inverse document frequency) score.

In addition, metadata is a very important feature to integrate especially when we deal with non textual information such as images.

In order to use these features for the similarity computation we select appropriate feature weights. We choose to use the standard retrieval metric tf.idf as keyphrase weight as it is commonly used in IR. Regarding metadata features, we assume as in [14] that appropriate weights are given to the metadata by the server maintainer. Tf.idf scores combined to metadata weights are then properly normalized.

Given a server document *sd*, its vector representation is given by:

$$Matrix(sd) = \langle w(f_1,sd), w(f_2,sd), \ldots, w(f_i,sd), \ldots, w(f_k,sd) \rangle$$

Where $w(f_i,sd)$ corresponds to the weight of the feature f_i in document *sd*; k being the total number of features extracted from the server. Document vectors are stored in an inverted file containing a dictionary of all extracted features and posting files for each feature specifying the documents in which it appears and its corresponding weights. We also output a file for which each entry contains a document id along with the list of features that have been extracted from the document.

4.3 Usage preprocessing

This part includes the standard steps followed in usage preprocessing, namely data cleaning, and identification of users, user sessions, user transactions, and path completion [5, 14]. Usage preprocessing in UserWatcher is data centric as the objective is to retrieve the part of the server data that has been accessed by the user and then search the Web for user related documents. We adapt our work in transaction log analysis [6, 13] to integrate new solutions that have been proposed in usage preprocessing.

As part of data cleaning we identify the data that is relevant to our objectives namely flat data, metadata. Document data is not explicitly stored in the usage content. Regarding metadata extracted from Web pages, we consider only the hyperlinks. Hyperlink labels will be added to the list of flat data. Hyperlink references are used to retrieve the document ids.

User identification in UserWatcher needs to achieve two objectives: (1) being able to distinguish the actions performed by each user and (2) being able to explicitly identify a user in the Web community. Regarding the first objective we integrate the heuristics proposed in [5] to enhance the capabilities of the user identification system that we used in [6, 13]. We also use the approach presented in [18] to detect reboot actions and filter them out from the normal class of users. Achieving the second objective relies on the availability of data gathered from the user and stored in files such as user profiles and data registry. Data gathered for identifying users varies from simple information about the domain of the machine used to connect to the server to a more detailed description of the users identity (name, address, etc.).

Once the users have been identified, the sequence of actions performed by each user needs to be broken down into sessions. As in [6, 13], we assume that a session is a set of actions performed by the same user with no more than thirty minutes lapse between two actions. Finally, we consider a user transaction as the set of all actions performed within a session; that is a user transaction corresponds to a user session.

Each user transaction is split into two parts and stored into two separated transaction files. The first file deals only with flat data, including hyperlink labels, that have been accessed during user sessions. In the second file, user session entries include identifiers of server documents that have been accessed during the sessions. Information in each file will be processed differently in the next phase: pattern discovery.

5 Pattern discovery

Data output from server content preprocessing and usage content preprocessing are used in the discovery process. Recall that the objective is to search for users that produce products presenting some similarity with the server products. The notion of products is very broad. Example of products include research techniques/results, Web advertisement strategy, techniques in Web pages design, etc. We assume that this information can be represented in one of the types of data presented in earlier sections.

Three main steps are performed in this phase. First, locating user related documents, then preprocessing them to extract documents' features; and finally perform similarity computation between users' documents and server's documents that have been accessed by the user. Interesting patterns are output as a set of recommendations for the decision maker.

5.1 Locating user documents

Transaction file output from the usage preprocessing step includes two main types of data: *user's identification data and server data*.

Server data is either flat data or a list of document ids accessed by the user. Document ids are combined with the results of the server content preprocessing to retrieve only the part of the inverted file corresponding to the documents accessed by the user. This subset is referred to as "user's related subset" in figure 1.

Features extracted from *user's related subset* are added to user flat data. Duplicate information is eliminated. Combined with user identification data, the information is exploited to locate documents from the publicly indexable Web.

For this process, we use multiple search engines such as Altavista and Google; and we derive multiple queries to tailor the searching features of each Web crawler. As an example, keyphrases including more than word are the target of search engines such as Altavista; while keyphrases including one word combined with user identification are quickly processed by ResearchIndex. The results returned from each search engine are combined and duplicated are eliminated. Documents associated to the query results are stored temporarily in the server for further processing. Documents that are Web pages are processed recursively to get all accessible pages using the hyperlink references. The resulting documents are referred to as user's retrieved documents in figure 1.

5.2 User's retrieved documents analysis

User's retrieved documents are processed in a similar way as we did for server content documents. For each document, we extract the first *five* keyphrases. Metadata is reduced to the hyperlink references as it is difficult to obtain the metadata describing the non-textual information. Features that are not part of the server vocabulary are discarded as the purpose is to find only user documents that are relevant to the server content. As a result, we associate to each user's retrieved document a set of features as document descriptors. The next step consists of attributing weights to document features. As the number of user retrieved documents is generally small compared to the number of server's documents, features weights of user's documents will be computed using server's documents.

We use tf.idf scores as feature weights; and we associate to each user document *ud* its vector representation given by:

$$\text{Matrix}(ud) = \langle w(f_1,ud), w(f_2,ud), \ldots, w(f_i,ud), \ldots, w(f_k,ud) \rangle$$

Where $w(f_i,ud)$ corresponds to the weight of the feature f_i in document *ud*. Note that k is the total number of features extracted from the server.

5.3 Similarity computation

The final step in the mining process consists of measuring the similarity between each user's retrieved document and each of the server document accessed by the user. We use the commonly used similarity measure known as the cosine measure [17]. Precisely, the similarity between a user retrieved document ***ud*** and a server document ***sd*** is given by:

$$\text{Sim}(sd, ud) = \frac{\sum_{i=1}^{k} w(fi, sd) \times w(fi, ud)}{\sqrt{\sum_{i=1}^{k} w^2(fi, sd) \times \sum_{i=1}^{k} w^2(fi, ud)}}$$

Score values exceeding some threshold value are selected. User retrieved documents associated to these scores are output. This is referred to as recommendation file in figure 1.

6 Conclusion

We have presented UserWatcher, a Web mining tool that integrates Web content mining and Web Usage Mining to monitor users accesses to web servers in order to detect whether the data accessed by a user is used to product similar information for the account of the user. The discovered information might indicate weaknesses in the system security policies and would trigger the review of user privileges.

UserWatcher exploits previous techniques that we developed in Web usage mining and Web content mining. The integration of the two components does not seem to present any major problem as it will be confirmed in the implementation phase. We plan to conduct a performance evaluation study to assess the effectiveness of the documents features. One particular extension that we would like to integrate is to have the system user define different levels of sensitivity for the server data; this will contribute into the computation of the features' weights. Based on the information provided by the decision maker, the system will automatically compute the new feature weights and perform another round of mining which will hopefully yield to improved results.

7 References

[1] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, V. Verykios. Disclosure Limitation of Sensitive Rules. Proc. of IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99), pp.45-52, 1999.

[2] S. Chakrabarti, et al. Mining the Link Structure of the World Wide Web. IEE Computer, 32(8):60-67, 1999.

[3] C. Clifton. Protecting Against Data Mining through Samples. In Proc. of 13th IFIP WG11.3 Conference on Database Security, Seattle, Washington, 1999.

[4] R. Cooley, B. Mobasher, J. Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web. ICTAI 1997.

[5] R. Cooley, et al. Data Preparation for Mining World Wide Web Browsing Patterns. Journal of Knowledge and Information Systems, (1) 1, 1999.

[6] S.J. Cunningham, M. Mahoui. Search Behavior in Two Digital Libraries: A Comparative Transaction Log Analysis. Proc. of the 3rd Int. Conference on Asian Digital Libraries ICADL2000, Seoul, Korea, pp 193-200, 2000.

[7] E. Frank, G. Paynter, I. Witten, C. Gutwin, C. Nevill-Manning. Domain-Specific Keyphrase Extraction. Proc. of the 16th Int. Joint Conference on Artificial Intelligence, Morgan-Kaufmann, pp 668-673, 1999.

[8] D. Gibson, J. Kleinberg, P. Raghavan. Inferring Web communities from Link Topology. Proc. of the 9th ACM Conference on Hypertext and Hypermedia, 1998.

[9] A. Herzberg, Y. Mass, J. Mihaeli. Access Control Meets Public Key Infrastructure. Proc. of IEEE Symposium on Security and Privacy, pp 2-14, 2000.

[10] S. Jones, M. Mahoui. Hierarchical Document Clustering Using Automatically Extracted Keyphrases. Proc. of 3rd Int.l Conference on Asian Digital Libraries ICADL2000, Seoul, Korea, 113-120, 2000.

[11] R. Kosala, H. Blockee. Web Mining Research: A Survey. SIGKDD Explorations, July 2000.

[12] Y. Maarek, I.Z. Ben Shaul, Automatically Organizing Bookmarks per Contents. Journal of Computer Networks and ISDN Systems, 28(7–11), p. 1321, 1997.

[13] M. Mahoui, S.J. Cunningham. A Comparative Transaction Log Analysis of Two Computing Collections. Proc. of the 4th European Conference on Research and Advanced Technology for Digital Libraries ECDL2000, September 2000.

[14] B. Mobasher, et al. Integrating Web Usage and Content Mining for More Effective Personalization. EC-Web 2000.

[15].M. Mohania, V. Kumar, Y. Kambayashi, B. Bhargava. Secured Web access. Proc. of Kyoto Int. Conference on Digital Libraries: Research and Practice, 2000.

[16] G. W. Paynter, I. H. Witten, and S. J. Cunningham. Evaluating Extracted Phrases and Extending Thesauri. Proc. of the 3rd Int. Conference on Asian Digital Libraries. Seoul, Korea, 2000.

[17] G. Salton. Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, 1989.

[18] P. Tan, V. Kumar. Discovery of Web Robot Sessions based on their Navigational Patterns. Technical Report, University of Minnesota, 2001.

[19] Y. Zhong, B. Bhargava. Authorization on Web Access, CERIAS TR 2001-17, Purdue University, 2001.