Detection of Sensitive Data and Information Leakage in a Trusted System

Ms. Reshma K Rajan[1] Mrs. Sijimol A S[2]
[1]M.Tech Student [2]Assistant Professor
[1,2]Department of Computer Science Engineering
[1,2]MBCCET Peermade, Idukki, Kerala, India

Abstract— Data-leak issues have developed quickly. Among different information release cases, human missteps are one of the primary drivers of information loss. It contains a Privacy Preserving Data-Leak Detection (DLD) solution to solve the data leak issue where it scans the characters of the pattern from right to left beginning with the rightmost one. Detection of sensitive data and information leakage in a trusted system is possible using Boyer Moore algorithm. Boyer Moore algorithm allows supporting the pattern matching. Proposes the Hash to Hash Value Comparison concept which lowers the execution time of string with the help of hash to hash match. First the hash value of the string corresponding to the content has been calculated. Then split up the string, word by word or as per the wish and taking the hash value of that split up string and then it compares with the previous hash value. If the hash values match with each other then the alert message is passed from the DLD Provider which detects the file duplication or file leakage to the data owner and the data user. This method detects the false alarm rate and reduce the execution time while it comparing with Boyer Moore Algorithm.

Key words: DLD Provider, Data duplication, File Leakage Detection

I. Introduction

As considering the statistics of business firms and research organizations data duplication over the network have grown rapidly in recent years. So the data leakage is happening while the data undergoes to be duplicated. Human mistakes are one of the main causes of data leakage problem. Therefore the data leak detection must be necessary to solve the issue. Privacy Preserving data leak detection aims to find the data duplication where the data owner and data user are involved. When a user is uploading a file, the same file can be downloaded and uploaded by the other users, then this file is said to be in duplication. Boyer Moore Algorithm is used for the pattern matching of the contents which is present in each of the data files. Boyer Moore Algorithm is an efficient pattern matching algorithm where it scans the characters of the pattern from right to left beginning with the rightmost one instead of Naïve method.

Detection of sensitive data and information leakage in a trusted system has been done using Boyer Moore algorithm. Boyer Moore algorithm allows supporting the pattern matching. Proposes the Hash to Hash Value Comparison concept along with the use of Boyer Moore Algorithm which lowers the execution time of string by the help of hash to hash match. First the hash value of the string corresponding to the content has been calculated. Then split up the string, word by word or as per the wish and taking the hash value of that split up string and then it compares with the previous hash value. If the hash values match with each other then the alert message is passed from the DLD Provider which detects the file duplication or file leakage to the data owner and the data user. DLD Provider provides the alert message to the corresponding data owner and data user who was already uploaded the file which detects to be same. It describes a privacy-preserving data-leak detection model for preventing data leak in trusted system. Privacy Preserving DLD model supports detection operation. Fuzzy Fingerprint technique is used for data leak detection where special data contents provided by the data owner from the DLD Provider [1]. Rabin fingerprint algorithm plays an important for preparing the fuzzy fingerprints from the contents of the corresponding files. Data owner can preprocess and prepare the fuzzy fingerprints and release it into the DLD Provider. The DLD Provider should monitor outbound traffic and detect in a proper manner. Then it should report all the data leaks.

II. Related work

Our work is most related to file leakage, and can break down the duplication of information documents led with our work. Privacy Preserving DLD assumes an indispensable part for the recognition of information spillage in a confided in framework. Security saving information spill identification technique goes about as an administration and limits the learning that a DLD supplier may accomplishes amid the procedure. Fundamentally there are the six activities performed by the information proprietor and the DLD supplier. This incorporate PREPROCESS dealt with by the information proprietor to set up the fingerprints of touchy information. These reviews of information RELEASE from the information proprietor specifically to the DLD supplier. At that point MONITORS and DETECTS happens in the DLD Provider. Report every one of the information spills on the off chance that it is influenced in the network [1]. Information security can be upgraded by Fuzzy fingerprint procedure amid information spill recognition activities. Our approach depends on a quick and reasonable one-route calculation on the delicate information, for example, SSN records, classified archives, touchy messages, and so on. Content-examination assignment should be possible for the advantage of information proprietor where safely designate the substance undertaking to DLD suppliers without uncovering the touchy information.

The point of the file leakage discovery is filtering content which is utilized as capacity and transmission for uncovered touchy information. Due to the vast substance and information volume, such a screening calculation should be adaptable for a convenient identification. Our answer utilizes the Map Reduce system for recognizing uncovered touchy substance, since it can self-assertively scale and use open assets for the errand, for example, Amazon EC2. Plan new Map Reduce calculations for registering accumulation convergence for information spill discovery. Our model actualized with the Hadoop framework accomplishes 225

Mbps investigation throughput with 24 hubs. Our calculations bolster a helpful protection safeguarding information change. This change empowers the security protecting procedure to limit the introduction of touchy information amid the identification. This change underpins the protected outsourcing of the information spill discovery to untrusted Map Reduce and cloud suppliers. Present another approach for exactly evaluating data spill limit in arrange activity. As opposed to hunting down known delicate information a unimaginable errand in the general case, we plan to gauge and oblige its most extreme volume. This examination tends to the danger of a programmer or noxious insider separating touchy data from a system. He or she could endeavor to take information without being recognized by concealing it in the commotion of ordinary outbound activity. For web movement, this frequently implies reserving bytes in ways or header fields inside apparently considerate solicitations. To battle this danger, we misuse the way that an extensive part of honest to goodness organize movement is rehashed or obliged by convention details. This settled information can be disregarded, which segregates genuine data leaving a system, paying little respect to information concealing procedures. The break estimation procedures exhibited here spotlight on the Hypertext Transfer Protocol (HTTP), the fundamental convention for web perusing. They exploit HTTP and its connection with Hypertext Markup Language (HTML) reports and Javascript code to evaluate data spill limit.

III. Problem formulation

A. Problem Definition:

The Data leak detection technique allows finding out the leakage of confidential data. The Data owners contains the confidential data, it has to deliver to the authorised end user. But that data may be accidentally leaked or found in unauthorised user. Data leak detection technique identifies the leaked data. It ensures the efficiency of sensitive data. The main drawback with current privacy preserving DLD for detecting the leakage of sensitive file is getting more time by using an algorithm. Fuzzy Fingerprint technique takes the digests of the file and performs some of the operations and generates string corresponding to the file. Then put the string to the DLD Provider which detects whether the leakage is found or not. Thus efficiency will be poor in accordance with the current privacy preserving DLD.

B. Existing System:

Our security protecting information spill recognition technique underpins pragmatic information release identification as an administration and limits the learning that a DLD owner may pick up amid the procedure. There are six tasks executed by the data owner and the DLD provider in our convention. They incorporate PREPROCESS keep running by the information proprietor to set up the summaries of delicate information, RELEASE for the information proprietor to send the overviews to the DLD provider MONITOR and DETECT for the DLD supplier to gather active traffic of the association, process condensations of traffic content, and recognize potential breaks, REPORT for the DLD supplier to return information spill cautions to the information proprietor where there might be false positives (i.e., false alerts), and POSTPROCESS for the information proprietor to pinpoint genuine information spill occasions. Points of interest are exhibited in the following segment. The convention depends on deliberately processing information closeness, specifically the quantitative similitude between the delicate data and the watched arrange traffic. High closeness demonstrates potential information spill. For information spill identification, the capacity to endure a specific level of information change in traffic is vital. We allude to this property as commotion resistance. Our key thought for quick and clamor tolerant examination is the plan and utilization of an arrangement of nearby highlights that are delegates of neighborhood information designs, e.g., when byte b2 shows up in the touchy information, it is generally encompassed by bytes b1 and b3 shaping a neighbourhood design b1,b2,b3. Neighbourhood highlights save information designs notwithstanding when modifications (inclusion, erasure, and substitution) are made to parts of the information. For instance, if a byte b4 is embedded after b3, the neighborhood design b1, b2, b3 is held however the worldwide example (e.g., a hash of the whole report) is obliterated. To accomplish the protection objective, the information proprietor produces an uncommon kind of summaries, which we call fluffy fingerprints. Naturally, the reason for fluffy fingerprints is to conceal the genuine touchy information in a group. It keeps the DLD supplier from taking in its correct esteem.

The DLD Provider acquires reviews of delicate information from the data owner. The data owner utilizes a sliding window and Rabin fingerprint calculation [12] to create short and hard to-switch (i.e., one-route) processes through the quick polynomial modulus task. The sliding window creates little pieces of the prepared information (delicate information or system traffic), which saves the nearby highlights of the information and gives the commotion resilience property. Rabin fingerprints are figured as polynomial modulus tasks, and can be executed with quick XOR, move, and table look-into activities. The Rabin fingerprint calculation has an exceptional min-wise freedom property [11], which bolsters quick irregular fingerprints choice (in uniform distribution) for halfway fingerprints divulgence. The shingle-and-fingerprint process is defined as takes after. A sliding window is utilized to produce q-grams on an info paired string first. The fingerprints of q-grams are then processed. A shingle (q-gram) is a fixed-measure succession of coterminous bytes. For instance, the 3-gram shingle set of string abcdefgh comprises of six components {abc, bcd, cde, def, efg, fgh}. Nearby component conservation is expert using shingles. Thusly, our approach can endure delicate information modification to some degree, e.g., embedded labels, little measure of character substitution, and softly reformatted information. The utilization of shingles for finding copy web reports first showed up in [12] and [13].

IV. Proposed system

The inspiration for the proposed work is that, the execution time lowers by implementing the hash to hash comparison concept. A hash function maps a variable length input string to fixed length output string — its hash value, or hash for short. Along with the hash to hash comparison concept, the pattern matching algorithm is used for retrieving the content of the file. Boyer Moore Algorithm is used as the pattern matching algorithm where it scans the contents from the right to the left rather than using the Naïve method where it always scans from the left to right. Mainly there are two heuristic approach is used in Boyer Moore Algorithm such as Bad Match Table and Good suffix rule. Good Suffix Rule is somewhat difficult for the part of implementation. Mainly we consider the two important things before using the Boyer Moore Algorithm.

1) Compare Pattern to the text, starting from the rightmost character in patterns.
2) When the mismatch occurs shift the pattern to the right corresponding to the value in Bad Match Table.

Bad Match Table aims to compare the pattern to the text. The value can be computed by evaluating an equation. Value= Length-Index-1 is used as the value of each letter from the pattern for the bad match table. For an example, WELCOMETOTEAMMAST is the text and TEAMMAST is the pattern that we need to search whether the given pattern is present or not in the corresponding text mentioned above. The searching can be done by looking into the Bad Match Table where values are updated for each letter of the pattern. Here there are eight letters in the pattern TEAMMAST. So the length is considered as 8. Bad Match Table does not allow entering the repetition of same letter. In TEAMMAST where t, a, m are repeated, so it is permitted to enter once in a Bad Match Table. * symbol is used at the end of the pattern .The value of the symbol * is always considered as the length of the given pattern. Here the value of * is 8 .The Last computing value is updated in the Bad Match Table incase if the repetition of the same letter occurs. The first letter of the pattern is always saved the value as the length of the pattern. So here, TEAMMAST begins with T and the updation of the table takes the value of T as 8. After completing the updation of the table, we just compare the pattern to the text by looking into the values of the Bad Match Table. The searching criteria have been followed by comparing the letter of the pattern with the corresponding letter of the text from right to left. If the letters of both the pattern and text matches then moving to the next letter and similarly checks whether match or mismatch occurs. When the mismatch occurs shift the pattern to the right corresponding to the value in the Bad Match Table.

When a data owner has ready to upload the file, the hash value of the contents of that particular file can be computed. If the data users have been already download and upload the same file which can be owned by the data owner (admin) where the hash value of the contents of that file can be computed. Hash to Hash comparison concept along with the Boyer Moore which is used for comparing the contents of the files uploaded by the data owner and the data users that help to lowers the execution time of the string generated by the files. If the computed Hash values are same then the data duplication occurs and thereby data leakage of the files in a trusted system has happening there. The DLD Provider will give the alert to the data owner and the data user through sending an email if the duplication of the file occurs or when the computed hash values were same. If the hash value of the files uploaded by the data owner and the data user are different, then the leakage of the file could not exists.

The proposed concept always lowers the execution time and getting more efficient rather than using any algorithm. Hash functions can be considered for taking the hash value of the files in Hash to Hash comparison concept.

V. PROPOSED IMPLEMENTATION SCHEME

The Proposed Hash to Hash comparison concept always lowers the execution time. First the hash value of the string corresponding to the content has been calculated. Then split up the string, and taking the hash value of that split up string. After it compares with the previous hash value.

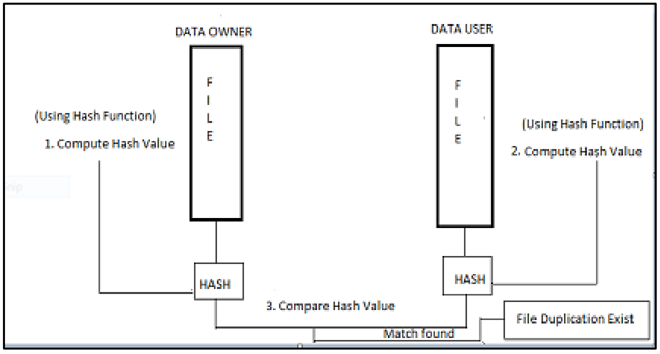

Fig. 1: Hash to Hash Comparison

If the hash values match with each other then the alert message is passed from the DLD Provider to the data owner and the data user. Hash capacities take contribution of sort Byte, it may be important to change over the source into a byte cluster before it is hashed. Proclaim a string variable to hold the source information, and two byte exhibits to hold the source bytes and the subsequent hash esteem. Contrast two byte clusters are with circle through the exhibits. Contrasting every individual component with its partner from the second esteem. On the off chance that any components are unique, at that point the two esteems are not equivalent. Boyer Moore Algorithm along with hash to hash comparison concept is used as the pattern matching algorithm where it scans the contents from the right to the left rather than using the Naïve method where it always scans from the left to right. Mainly there are two heuristic approach is used in Boyer Moore Algorithm such as Bad Match Table and Good suffix rule. Good Suffix Rule is somewhat difficult for the part of implementation. Proposed concept helps for the detection of the false alarm rates as well as it efficiently lowers the execution time. Hash functions are used which is inbuilt for comparing the contents of the file.

A hash work is any capacity that can be utilized to delineate of discretionary size to information of settled size. The qualities returned by a hash work are called hash esteems, hash codes, digests, or basically hashes. One utilize is an information structure called a hash table, generally utilized as a part of PC programming for fast information query. Hash capacities quicken table or database query by recognizing copied records in a huge document. Using these hash function the hash value of the file provided by the data owner can be taken. Then the hash value of the file that can be uploaded by the data user can be computed. Then only comparing both the files for the appropriate result that we need. Less execution time is needed for generating the string that correspond to the files. This advantage plays an important role while the detection of file leakage has been happened. Along with the Boyer Moore Algorithm, Hash to Hash comparison concept achieves more efficiency which leads to the proper execution time to generate the string and also helps to detect false alarm rates.

VI. Conclusion

This paper considers the detection of sensitive data leakage in a trusted system. The proposed concept is Hash to Hash Comparison mechanism which always lowers the execution time of the string that corresponds to the contents of the file. If one of the file can be uploaded by the data user then the other file which can be provided by the data user. Privacy Preserving DLD ensures whether both the files that can be uploaded by the data owner and the data user are same. This can be detected by comparing the hash values of the string that can be generated as per the files along with the pattern matching algorithm like Boyer Moore Algorithm. It is an efficient pattern matching algorithm which scans the contents of the file from right to left instead of Naïve Method. Comparing the hash value of both files always aims to get the execution faster rather than using any other algorithms.

References

[1] Xiaokui Shu, Danfeng Yao, and Elisa Bertino, “Privacy Preserving Detection of Sensitive Data Exposure” IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 5, MAY 2015.

[2] Cantone, D. and Faro, S. 2003. Fast-Search: a new efficient variant of the Boyer-Moore string matching algorithm. In WEA 2003. Lecture Notes in Computer Science, vol. 2647. Springer-Verlag, Berlin, 247–267.

[3] X. Shu and D. Yao, “Data leak detection as a service,” in Proc. 8th Int. Conf. Secur. Privacy Commun. Netw., 2012, pp. 222–240.

[4] Identity Finder. Discover Sensitive Data Prevent Breaches DLP Data Loss Prevention. [Online]Available:http://www.identityfinder.com/, accessed Oct. 2014.

[5] K. Borders and A. Prakash, “Quantifying information leaks in outbound web traffic,” in Proc. 30th IEEE Symp. Secur. Privacy, May 2009, pp. 129–140.

[6] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, “Panorama: Capturing system-wide information flow for malware detection and analysis,” in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 116–127.

[7] K. Borders, E.V. Weele, B. Lau, and A. Prakash, “Protecting confidential data on personal computers with storage capsules,” in Proc. 18th USENIX Secur. Symp., 2009, pp. 367–382.

[8] A. Nadkarni and W. Enck, “Preventing accidental data disclosure in modern operating systems,” in Proc. 20th ACM Conf. Comput. Commun. Secur., 2013, pp. 1029–1042.

[9] A. Kapravelos, Y. Shoshitaishvili, M. Cova, C. Kruegel, and G. Vigna, “Revolver: An automated approach to the detection of evasiveweb-based malware,” in Proc. 22nd USENIX Secur. Symp., 2013, pp. 637–652.

[10] X. Jiang, X. Wang, and D. Xu, “Stealthy malware detection and monitoring through VMM-based ‘out-of-the-box’ semantic view reconstruction,” ACM Trans. Inf. Syst. Secur., vol. 13, no. 2, 2010, p. 12.

[11] G. Karjoth and M. Schunter, “A privacy policy model for enterprises,” in Proc. 15th IEEE Comput. Secur. Found. Workshop, Jun. 2002, pp. 271–281.

[12] A. Z. Broder, “Some applications of Rabin’s fingerprinting method,” in Sequences II. New York, NY, USA: Springer-Verlag, 1993, pp. 143–152.

[13] Z. Broder, “Identifying and filtering near-duplicate documents,” in Proc. 11th Annu. Symp. Combinat. Pattern Matching, 2000, pp. 1–10.

[14] Wang, S. Yu, W. Lou, and Y. T. Hou, “Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud,” in Proc. 33th IEEE Conf. Comput. Commun., Apr./May 2014, pp. 2112–2120.

[15] Constantine P. Sapuntzakis, Ramesh Chandra, Ben Pfaff, Jim Chow, Monica S. Lam, and Mendel Rosenblum. Optimizing the migration of virtual computers. In Proceedings of the 5th Symposium on Operating Systems Design and Implementation, 2002.