International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 4 380 – 384

Survey on Data Leak Detection of Sensitive Data Exposure for Preserving Privacy

Avanti M. Ganorkar
P. R. Pote (Patil) Welfare & Education Trust's college of Engineering & Management, Amravati
Department of Computer Science & Engineering
Amravati, India
ganorkaravanti@gmail.com

Prof. Praful B. Sambhare
P. R. Pote (Patil) Welfare & Education Trust's college of Engineering & Management, Amravati
Department of Computer Science & Engineering
Amravati, India
Sambharepraful832@gmail.com

Abstract— Now-a-days large amount of data leaks occur in various research institutions, organization and security firms. The data leakage occurs due to the improper protection to the data. Deliberately planned attacks, inadvertent leaks (e.g. forwarding confidential emails to unclassified email accounts), and human mistakes (e.g. assigning the wrong privilege) lead to most of the data-leak incidents .The common way is used to monitor the data that are stored in a organizational local network. However, this requirement is undesirable, as it may threaten the confidentiality of the sensitive information .For existing method we require plaintext sensitive data. A privacy preserving data-leak detection solution is proposed which can be outsourced and be deployed in a semi-honest detection environment. In this paper, fuzzy fingerprint technique is designed and implemented to enhance data privacy during data leak detection operation. The DLD provider computes fingerprints from network traffic and identifies potential leaks in them. The estimation result shows that this method can provide accurate detection.

Keywords- *Data Leak, Network Security, Privacy, Collection Intersection.*

*****

I. INTRODUCTION

Detecting and preventing data leaks requires a set of complementary solutions, which may include data-leak detection, data confinement [2][3], stealthy malware detection and policy enforcement. Network data-leak detection (DLD) typically searches for any occurrences of sensitive data patterns and performs deep packet inspection (DPI). DPI is a technique to analyze payloads of TCP/IP packet for inspecting application layer data, e.g., HTTP header/content. Alerts are triggered and traffic passes a threshold when the amount of sensitive data found. There are two types of input sequences in data-leak detection model: sensitive data sequences and content sequences. (1) Sensitive data contain the sensitive information that cannot be exposed to unauthorized parties, e.g., proprietary documents, customers' records. Sensitive data can also be partitioned to small sensitive data sequence. (2) Content is the data to be inspected occurrences of sensitive data patterns. The detection need to partition the original content stream into content segments [1]. The detection system can be deployed on a router or integrated into existing network intrusion detection systems (NIDS). Straightforward realizations of data-leak detection require the plaintext sensitive data. However, this requirement is undesirable, as it may threaten the confidentiality of the sensitive information. If a detection system is compromised, then it may expose the plaintext sensitive data (in memory). In addition, the data owner may need to outsource the data-leak detection to providers, but may be unwilling to reveal the plaintext sensitive data to them. Therefore, one needs new data-leak detection solutions that allow the providers to scan content for leaks without learning the sensitive information.

In this paper, we present details of our solution and provide extensive experimental evidences and theoretical analyses to demonstrate the feasibility and effectiveness of our approach.

Our contributions are summarized as follows.

1) We describe a privacy-preserving data-leak detection model for preventing inadvertent data leak in network traffic. Our model supports detection operation delegation and ISPs can provide data-leak detection as an add-on service to their customers using our model. We design, implement, and evaluate an efficient technique, fuzzy fingerprint, for privacy-preserving data-leak detection. Fuzzy fingerprints are special sensitive data digests prepared by the data owner for release to the DLD provider.

2) We implement our detection system and perform extensive experimental evaluation on 2.6 GB Enron dataset, Internet surfing traffic of 20 users, and also 5 simulated real-world data-leak scenarios to measure its privacy guarantee, detection rate and efficiency. Our results indicate high accuracy achieved by our underlying scheme with very low false positive rate. Our results also show that the detection accuracy does not degrade much when only partial (sampled) sensitive-data digests are used. In addition, we give an empirical analysis of our fuzzification as well as of the fairness of fingerprint partial disclosure.

To prevent the DLD provider from gathering exact knowledge about the sensitive data and the collection of potential leaks is composed of noises and real leaks. The data owner who post-processes the potential leaks sent back by the DLD provider and then determines whether there is any real data leak.

II. SYSTEM OVEREVIEW

The privacy-preserving data-leak detection problem with a threat model, a security goal and a privacy goal is abstracted. First we describe the two most important players in our abstract model: the organization (i.e., data owner) and the data-leak detection (DLD) provider.

•Organization owns the sensitive data and authorizes the DLD provider to inspect the network traffic from the organizational networks for anomalies, namely inadvertent data leak. However, the organization does not want to directly reveal the sensitive data to the provider.

•DLD provider inspects the network traffic for potential data leaks. The inspection can be performed offline without causing any real-time delay in routing the packets. However, the DLD provider may attempt to gain knowledge about the Sensitive data.

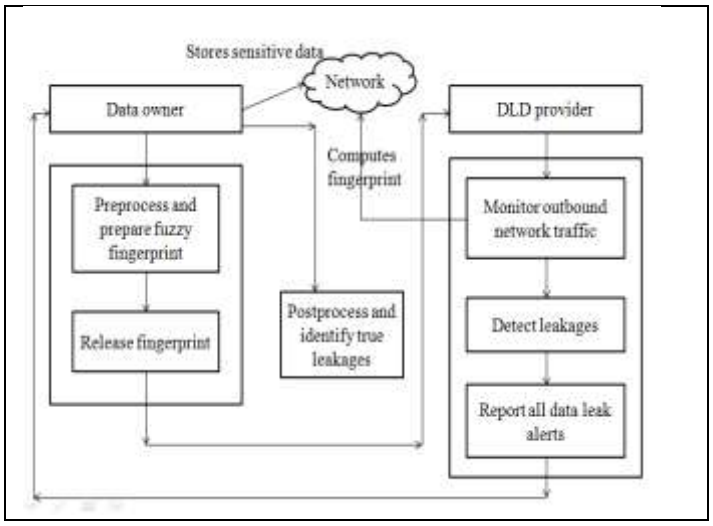

Fig.1: System Overview [4]

In our data leak detection model, we analyze two types of sequences: sensitive data sequence and content sequence.

• *Content sequence* is the sequence to be examined for leaks. The content may be data extracted from file systems on personal computers, workstations, and servers; or payloads extracted from supervised network channels.

• *Sensitive data sequence* contains the information such as customer's records, proprietary documents that needs to be protected and cannot be exposed to unauthorized parties. The sensitive data sequences are known to the analysis system.

In this paper, we focus on detecting inadvertent data leaks, and we assume the content in file system or network traffic (over supervised network channels) is available to the inspection system. A supervised network channel could be an unencrypted channel or an encrypted channel where the content in it can be extracted and checked by an authority. Such a channel is widely used for advanced NIDS where MITM (man-in-the-middle) SSL sessions are established instead of normal SSL sessions [5]. We do not aim at detecting stealthy data leaks that an attacker encrypts the sensitive data secretly before leaking it. Preventing intentional or malicious data leak ,especially encrypted leaks, requires different approaches and remains an active research problem [6].In our current security model, we assume that the analysis

System is secure and trustworthy. Privacy-preserving data-leak detection can be achieved by leveraging special protocols and computation steps [7]. It is another functionality of a detection system, and the discussion is not within the scope of this paper.

A. Technical Challenges

1) High Detection Specificity: In our data-leak detection model, high specificity refers to the ability to distinguish true leaks from coincidental matches. *Coincidental matches* are false positives, which may lead to false alarms. Existing set-based detection is orderless, where the order of matched shingles (*n*-grams) is ignored. Orderless detection may generate coincidental matches, and thus having a lower accuracy of the detection. In comparison, our alignment-based method has high specificity. For example, a detection system can use 3-grams to represent the sensitive data.

Sensitive data `abcdefg`

3-grams `abc`, `bcd`, `cde`, `def`, `efg`

Then, consider the content streams 1 and 2 below.

Stream 1 contains a true leak, whereas stream 2 does not.

Content stream 1 `....abcdefg...`

Content stream 2 `....efg...cde...abc...`

However, set intersection between 3-grams of the sensitive data and the 3-grams of content stream 2 results in a significant number of matching 3-grams (`efg`, `cde`, and `abc`), even though they are out of order compared to the sensitive data pattern. This problem is eliminated in alignment, i.e., the content stream 2 receives a low sensitivity score when aligned against the sensitive data.

2) Pervasive and Localized Modification: Sensitive data could be modified before it is leaked out. The modification can occur throughout a sequence (pervasive modification). The modification can also only affect a local region (local modification). We describe some modification examples:

• Character replacement, e.g., `Word Press` replaces every space character with a **+** in HTTP POST requests.

• String insertion, e.g., HTML tags inserted throughout a document for formatting or embedding objects.

• Data truncation or partial data leak, e.g., one page of a two-page sensitive document is transmitted.

B. Overview of Our Approach

Our work presents an efficient sequence comparison technique needed for analyzing a large amount of content for sensitive data exposure. Our detection approach consists of a comparable sampling algorithm and a sampling oblivious alignment algorithm. The pair of algorithms computes a quantitative similarity score between the sensitive data a quantitative similarity score between the sensitive data and the content. Local alignment – as opposed to global alignment [5] – is used to identify similar sequence segments. The design enables the detection of partial data leaks. Our detection runs on continuous sequences of *n* bytes (*n*-grams). *n*-grams are obtained from the content and sensitive data, respectively. Local alignment is performed between the two (sampled) sequences to compute their similarity. The purpose of our comparable sampling operation is to enhance the analysis throughput. We discuss the tradeoff between security and performance related to sampling in our evaluation sections. Finally, we report the content that bears higher-than threshold similarity with respect to sensitive patterns. Given a threshold *T* , content with a greater-than –T sensitivity is reported as a leak.

III. TECHNIQUES USED IN PRESERVING PRIVACY OF SENSITIVE DATA

A. Fuzzy fingerprint method and protocol

The technical details of our fuzzy fingerprint mechanism are described below.

A. Shingles and Fingerprints

The DLD provider obtains digests of sensitive data from the data owner. Data owner uses a sliding window and Rabin fingerprint algorithm [9] to generate short and hard to- reverse digests through the fast polynomial modulus operation. The sliding window generates small fragments of the processed data (sensitive data or network traffic), and which preserves the local features of the data and provides the noise tolerance property. The Rabin fingerprint algorithm has a unique min-wise independence property [8], which supports fast random fingerprints selection (in uniform distribution) for partial fingerprints disclosure. Rabin fingerprints are computed as polynomial modulus operations, and can be implemented with fast XOR, shift, and table look-up operations. The shingle-and-fingerprint process is defined as - A sliding window is used to generate first q-grams on an input binary string. The fingerprints of q-grams are then computed. A shingle (q-gram) is a fixed-size sequence of contiguous byte. For example, 3-gram shingle set of string abcdefgh consists of six elements {abc, bcd, cde, def, efg, fgh}.The Local feature preservation is accomplish through the use of shingles. Therefore, our app1roach can tolerate sensitive data modification to some extents, e.g., lightly reformatted data, small amount of character substitution, and inserted tags. The use of shingles alone does not satisfy the one-wayness requirement. Rabin fingerprint is utilized to satisfy such requirement after shingling. The process of fingerprinting, each shingle is treated as a polynomial q(x). Each coefficient of q(x), i.e., ci($0 < i < k$) is one bit in the shingle and q(x) is mod by a selected irreducible polynomial p(x). The process shown in (1) maps a k-bit shingle into a pf -bit fingerprint f where the degree of p(x) is p f + 1.

$f = c1xk-1 + c2xk-2 + \ldots + ck-1x + ck \bmod p(x)$ (1)

For the detection perspective, a straightforward method is for the DLD provider to raise an alert if any sensitive fingerprint matched by the fingerprints from the traffic. However, this approach has a privacy issue. In any case, if there is a data leak and there is a match between two fingerprints from sensitive data and network traffic. Then, the DLD provider learn the corresponding shingle as it knows the content of the packet[1].

Definition: Given a pf -bit-long fingerprint f, the fuzzy length pd (pd < pf) is the number of bits in f that may be perturbed by the data owner[1].

B. Operations in Protocol

1) *preprocess:* The preprocess operation is run by the data owner on each piece of sensitive data [1].

a) The data owner chooses four public parameters ((q, p(x), pd, M). q is the length of a shingle. p(x), is an irreducible polynomial (degree of pf + 1) used in Rabin fingerprint. Each fingerprint is pf –bit long and the fuzzy length is pd . M is a bitmask, which is p f -bit long and contains pd 0‟s at random positions. The positions of 1‟s and 0‟s in M indicate the bits to preserve and to randomize in the fuzzification, respectively.

b) The data owner computes S which is the set of all Rabin fingerprint of the piece of sensitive data.

c) The data owner transforms each fingerprint f ∈ S into a fuzzy fingerprint f * with randomized bits (specified by the mask M). The procedure is described as follows- for each f ∈ S and the data owner generates a random pf -bit binary string f`, mask out the bits not randomized by f`‟ = (NOT M) AND f` and fuzzify f with f * = f XOR f`‟. The overall computation is described in equation(2).

f * = ((NOT M) AND f`) XOR f (2)

All fuzzy fingerprints are collected and then it form the output of this operations as the fuzzy fingerprint set, S*.

2) *release:* This operation is run by the data owner. The fuzzy fingerprint set S* obtained by PREPROCESS is released to the DLD provider for use in the detection, along with the public parameters (q, p(x), pd, M). Data owner keeps S for use in the subsequent POSTPROCESS operation [1].

3) *monitor:* The monitor operation is run by the DLD provider. DLD provider monitors the network traffic T from the data owners organization. The entire packet in T is collected and the payload of it is sent to the next operation as the network

traffic (binary) string T˜. The payload of each packet is not the only choice to define T˜. A more sophisticated approach could identify TCP flows and then extract contents in a TCP session as ˜T.

4) *detect*: This operation is run by the DLD provider on each T˜ as described below.

(a) The DLD provider first computes the set of Rabin fingerprints of traffic content T˜ based on the public parameter. This set is denoted as T.

(b) The DLD provider tests whether each fingerprint f „∈ T is also in S∗ using the fuzzy equivalence test in equation (3).

E (f ‘, f ∗) = NOT (M AND (f ‘XOR f ∗)) (3)

E (f„, f∗) is either True or False and, f XOR f∗ gives the difference between f „ and f ∗. M AND (f‟ XOR f∗) filters the result leaving only the interesting bits. Because XOR yields 0 for equivalent bits and NOT is used to turn 0-bits into 1‟s (and 1‟s into 0‟s). The overall result from (3) is read as a Boolean indicating whether or not f‟ is equivalent to a fuzzy fingerprints f ∗ ∈ S∗. (2) and (3) are designed in a pair, and M works the same in both equations by masking out fuzzified bits at same positions in each f , f ∗ and f „. All f„ with True values are recorded in a set ˆT .

(c) DLD provider aggregates the outputs from the preceding step and raises alerts based on a threshold.

5) *report*: If DETECTION on T˜ yields an alert the DLD provider reports the set of detected candidate leak instances ˆT to the data owner.

6) *postprocess*: After receiving ˆT and the data owner test every f „∈ ˆT to see whether it is in S.

- *Rabin fingerprint algorithm*

This algorithm helps in generating a fuzzy fingerprint. It is used only after shingle is generated. generate digests of sensitive data through a one-way function, and then hide the sensitive values among other non-sensitive values via fuzzification. Using the min-wise independent property of Rabin fingerprint, the data owner can quickly disclose partial fuzzy fingerprints to the DLD provider. The purpose of partial disclosure is two-fold: i) to increase the scalability of the comparison in the DETECT operation, and ii) to reduce the exposure of data to the DLD provider for privacy.

```
begin
int  prime = 101;
for each text
for (int i = 0; i < text.Length; i++)
char c = text[i];
hash1=c*(int)(Math.Pow(prime, text.Length - 1 -
i));
fingerprint=hash1;
end for;
```

```
end for;
```

I. END; RELATED WORK

In this paper, the privacy needs in an outsourced data-leak detection service and provide a systematic solution to enable privacy-preserving DLD services are identified. Shingle with Rabin fingerprint was used previously for identifying similar spam messages in a collaborative setting, as well as collaborative worm containment, virus scan, and fragment detection. Most data-leak detection products offered by the industry, e.g., Symantec DLP, Global Velocity identity Finder, do not have the privacy-preserving feature and cannot be outsourced. GoCloudDLP is a little different, which allows its customers to outsource the detection to a fully honest DLD provider. Fuzzy fingerprint method differs from these solutions and enables its adopter to provide data-leak detection as a service. The customer or data owner does not need to fully trust the DLD provider using our approach. Besides fuzzy fingerprint solution for data -leak detection, there are other privacy-preserving techniques invented for specific processes, e.g., DNA matching, or for general purpose use, e.g., secure multi-party computation (SMC). Similar to string matching methods discussed above, uses anonymous automata to perform comparison. SMC is a cryptographic mechanism, which supports a wide range of fundamental arithmetic, set, and string operations as well as complex functions such as knapsack computation, automated trouble-shooting , network event statistics, private information retrieval genomic computation, private database query, private join operations, and distributed data mining. The provable privacy guarantees offered by SMC comes at a cost in terms of computational complexity and realization difficulty. The advantage of fuzzy fingerprint approach is its concision and efficiency. Rabin fingerprint was used previously for identifying similar spam messages in a collaborative setting as well as collaborative worm containment virus scan and fragment detection In comparison, we tackle the unique data-leak detection problem in an outsourced setting where the DLD provider is not fully trusted. Such privacy requirement does not exist in above models, e.g., the virus signatures are non-sensitive in the virus-scan paradigm .We propose the fuzzy fingerprint approach to meet the special privacy requirement and present the first systematic solution to privacy-preserving data-leak detection with convincing results[9].

IV. RELATED WORK

There have been several advances in understanding the privacy requirement of security applications [12] or the Privacy needs [11]. Privacy preserving data-leak detection solution to solve the issue where a special set of sensitive data

digests is used in detection. The advantage of this method is that it enables the data owner to safely delegate the detection operation to a semi honest provider without revealing the sensitive data to the provider [1]. The aim of this this technique is finding the malicious users and detect the leak source effectively [10]. Goal is to identify the guilty agent when distributor's sensitive data have been leaked by some agents. In comparison, our work inspects traffic for signatures of sensitive-data and does not require any assumption on the patterns of normal header fields or payload. Furthermore, our solution provides privacy protection of the sensitive data against semi-honest DLD providers. We also give performance evidences indicating the efficiency of our solution in practice .The method of deep packet inspection is also widely used in network intrusion detection system (NIDS), such as SNORT [14] and Bro. They focus on designing and implementing efficient string matching algorithms [1] to handle short and flexible patterns in network traffic. However, NIDS is not designed for various kinds of sensitive data (e.g. long non-duplicated data), it may cause problems (e.g. large amount of states in an automata) in data leak detection scenarios. On the contrary, our solution is not limited to very special types of sensitive data, and we provide an unique privacy-preserving feature for service outsourcing. Besides fuzzy fingerprint solution for data -leak detection, there are other privacy-preserving techniques invented for specific processes, e.g., DNA matching, or for general purpose use, e.g., secure multi-party computation (SMC). Similar to string matching methods discussed above, uses anonymous automata to perform comparison. SMC is a cryptographic mechanism, which supports a wide range of fundamental arithmetic, set, and string operations as well as complex functions such as knapsack computation, automated trouble-shooting , network event statistics, private information retrieval genomic computation, private database query, private join operations, and distributed data mining. The provable privacy guarantees offered by SMC comes at a cost in terms of computational complexity and realization difficulty. The advantage of fuzzy fingerprint approach is its concision and efficiency.

V. CONCLUSION

This privacy preserving data-leak detection model make use of fuzzy fingerprint to find out the data leakages. Using special digests, the exposure of the sensitive data is kept to a minimum during the detection. We defined our privacy goal by quantifying and restricting the probability that the DLD provider identifies the exact value of the sensitive data. Our extensive experiments validate the accuracy, privacy, and efficiency of our solutions.

ACKNOWLEGEMENT

The author would especially grateful to guide Prof. Praful B. Sambhare and Prof. Vijay B. Gadicha Head of Computer Science and Engineering Department who has provided guidance, expertise and encouragement to complete this assignment.

REFERENCES

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

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

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

[4] International Journal of Recent Research in Mathematics Computer Science and Information Technology Vol. 2, Issue 2, pp: (121-126), Month: October 2015 – March 2016, Available at: www.paperpublications.org Page

[5] Y. Jang, S. P. Chung, B. D. Payne, and W. Lee, "Gyrus: A framework for user-intent monitoring of text-based networked applications," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 79–93.

[6] K. Borders and A. Prakash, "Quantifying information leaks in outboundnWeb traffic," in *Proc. 30th IEEE Symp. Secur. Privacy (SP)*, May 2009,pp. 129–140.

[7] S. Jha, L. Kruger, and V. Shmatikov, "Towards practical privacy for genomic computation," in *Proc. IEEE Symp. Secur. Privacy*, May 2008, pp. 216–230.s

[8] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," J. Comput. Syst. Sci., vol. 60, no. 3, pp. 630–659, 2000.

[9] X. Shu, J. Zhang, D. Yao, and W.-C. Feng, "Rapid and parallel content screening for detecting transformed data exposure," in *Proc. 3rd Int. Workshop Secur. Privacy Big Data (BigSecurity)*, Apr./May 2015, pp. 191–196.

[10] F. Hao, M. Kodialam, T. V. Lakshman, and H. Zhang, "Fast payload based flow estimation for traffic monitoring and network security," in *Proc. ACM Symp. Archit. Netw. Commun. Syst.*, Oct. 2005, pp. 211–220.

[11] J. Kleinberg, C. H. Papadimitriou, and P. Raghavan, "On the value of private information," in *Proc. 8th Conf Theoretical Aspects Rationality Knowl.*, 2001, pp. 249–257.

[12] S. Xu, "Collaborative attack vs. collaborative defense," in *Collaborative Computing: Networking, Applications and Worksharing* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 10. Berlin, Germany: Springer-Verlag, 200

[13] Yin Fan and Wang Lina , A Distribution Model for DataLeakage Prevention,2013 International Conference on Mechatronics Sciences, Electric Engineeringand Computer (MEC) Dec 20-22, 2013, Shenyang, China. 9, pp. 217–228.

[14] ROESCH, M. Snort: lightweight intrusion detection for networks In Proceedings of the 13th Conference on Systems Administration (LISA-99) (1999).