ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

An overview of confidential data leak detection method by preserving the privacy through Fuzzy Fingerprint method

Sachusha.S.Shaji[#1], Arun P Kuttappan[*2]
[#1] *Department of CSE, Gurudeva Institute Of Science & Technology ,Kottyam*
APJ Abdul Kalam Technological University
Kerala, India
[1] sachuja93@gmail.com
[2]arunkalesh@gmail.com

***Abstract*— Nowadays confidential datas are leaking in many public as well as the private firms. Many studies shows that confidential datas are leaking in many research institutions, government firms and in IT firms .Human errors or flaws are the main cause of the data leakage. Protecting the confidential data is a major concern for the organizations and individuals, where the confidential datas falls into fraudulent hands. So many organizations provide safety alert systems for detecting the data leakage of the system. In this paper, we present a data leak detection (DLD) solution to solve the issue where a special set of confidential data digests is used in detection. The main advantage of this method is that it enables the data owner to safely delegate the detection operation to a honest provider without revealing the confidential data to the provider. This paper also describes the details of fuzzy fingerprint mechanism for privacy preserving data-leak detection by randomized method for detection.**

***Index Terms*— Data leak, network security, privacy, collection intersection.**

I. INTRODUCTION

Data leakage has significantly growly in the recent years. Studies from many security firms, analysis institutions and government companies faces the problems of data leakage. Among those cases, human errors are considered has the most common data leak cause. Many researches focuses on the study of the data leak detection as many sensitive datas are losing because of human errors. Here in this paper we present a data leak detection solution to solve the issue where the special set of confidential data digests is used for the detection. We use fuzzy fingerprint method for detecting inadvertent data leak in network traffic. Network security consists of the policies adopted to prevent and monitor unauthorized access, misuse, modification, or denial of computer network and computer accessible resource. Its main feature is that the detection can be performed based on special digests without the sensitive data in plaintext, which minimizes the exposure of sensitive data during the detection.[2]

II. LITERATURE SURVEY

Based on the report from Risk Based Security (RBS), a large number of confidential data records have leaked dramatically during the last few years, i.e., from 412 million in 2012 to 822 million in 2013. Intentionally planned attacks, inadvertent leaks (e.g., forwarding confidential emails to unclassified email accounts), and human errors (e.g., assigning the wrong privilege) lead to most of the data-leak incidents [1].

The leak of confidential data either be it accidental or intentional, may cause huge losses to the data owner. Though there are number of systems designed for the data security by using different encryption algorithms, there is a big issue of the integrity of the users of those systems. It is very hard for any system administrator to trace out the data leaker among the system users. It creates a lot many ethical issues in the working environment.[3]

Most of the host-based solutions require the use of virtualization or special hardware to ensure the system integrity of the detector. This paper present a novel network-based data-leak detection (DLD) solution that is both efficient and privacy-preserving In comparison to host-based approaches, network-based data-leak detection focuses on analyzing the (unencrypted) content of outbound network packets for sensitive information.[4]

III. PROPOSED SYSTEM

In the proposed system, importance of fuzzy fingerprint method is mentioned.In the detection procedure, the data owner computes a special set of digests or fingerprints from the

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

sensitive data and then discloses only a small amount of them to the DLD provider. The DLD provider computes fingerprints from network traffic and identifies potential leaks in them. To prevent the DLD provider from gathering exact knowledge about the sensitive data, the collection of potential leaks is composed of real leaks and noises. It is the data owner, who post-processes the potential leaks sent back by the DLD provider and determines whether there is any real data leak.

Our privacy-preserving data-leak detection method supports practical data-leak detection as a service and minimizes the knowledge that a DLD provider may gain during the process. Fig. 1 lists the six operations executed by the data owner and the DLD provider in our protocol. They include PREPROCESS run by the data owner to prepare the digests of sensitive data, RELEASE for the data owner to send the digests to the DLD provider, MONITOR and DETECT for the DLD provider to collect outgoing traffic of the organization, compute digests of traffic content, and identify potential leaks, REPORT for the DLD provider to return data-leak alerts to the data owner where there may be false positives (i.e., false alarms), and POSTPROCESS for the data owner to pinpoint true data-leak instances. Details are presented in the next section. The protocol is based on strategically computing data similarity, specifically the quantitative similarity between the sensitive information and the observed network traffic. High similarity indicates potential data leak. For data-leak detection, the ability to tolerate a certain degree of data transformation in traffic is important. We refer to this property as *noise tolerance*. Our key idea for fast and noise-tolerant comparison is the design and use of a set of *local features* that are representatives of local data patterns, e.g., when byte $b2$ appears in the sensitive data, it is usually surrounded by bytes $b1$ and $b3$ forming a local pattern $b1, b2, b3$. Local features preserve data patterns even when modifications (insertion, deletion, and substitution) are made to parts of the data. For example, if a byte $b4$ is inserted after $b3$, the local pattern $b1, b2, b3$ is retained though the global pattern (e.g., a hash of the entire document) is destroyed. To achieve the privacy goal, the data owner generates a special type of digests, which we call fuzzy fingerprints. Intuitively, the purpose of fuzzy fingerprints is to hide the true sensitive data in a crowd. It prevents the DLD provider from learning its exact value[1].

FUZZY FINGERPRINT METHOD AND PROTOCOL

We describe technical details of our fuzzy fingerprint mechanism in this section.

A. Shingles and Fingerprints

The DLD provider obtains digests of sensitive data from the data owner.The data owner uses a sliding window and Rabin fingerprint algorihm [1] to generate short and hard to reverse (i.e., one-way) digests throgh the fast polynomial modulus operation. The sliding window geerates small fragments of the processed data (sensitive data or network traffic), which preserves the local features of the data and provides the noise tolerance property. Rabin fingerprints are computed as polynomial modulus operations, and can be implemented with fast XOR, shift, and table look-up operations. The Rabin fingerprint algorithm has a unique min-wise independence property [1], which supports fast random fingerprints selection (in uniform distribution) for partial fingerprints disclosure.

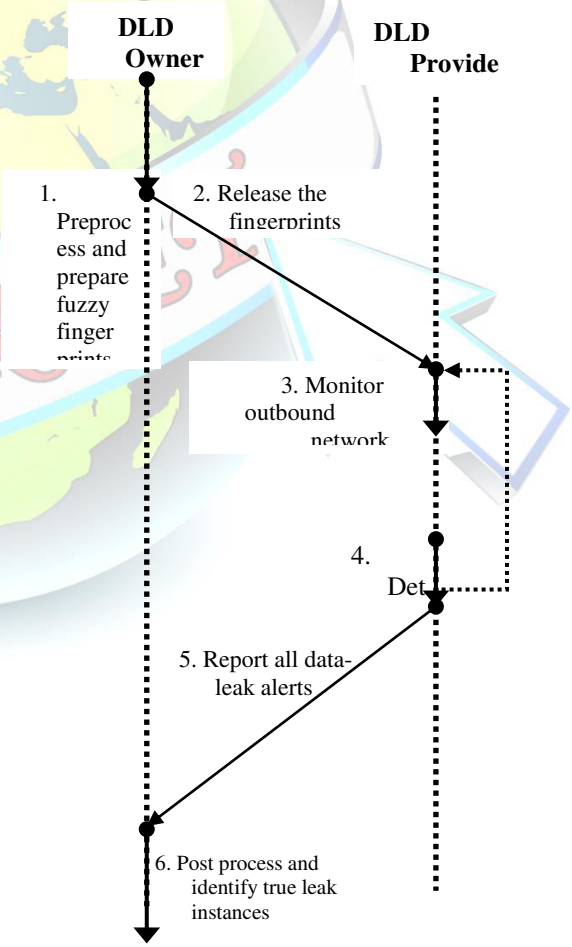

Fig.1 Model for the proposed system

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

The shingle-and-fingerprint process is defined as follows.
A sliding window is used to generate *q*-grams on an input
binary string first. The fingerprints of *q*-grams are then
computed.

Shingle (*q*-gram) is a fixed-size sequence of contiguous
bytes. For example, the 3-gram shingle set of string `abcdefgh` consists of six elements {`abc`, `bcd`, `cde`, `def`,`efg`, `fgh`}. Local feature preservation is accomplished through the use of shingles. Therefore, our approach can tolerate sensitive data modification to some extent, e.g., inserted tags, small amount of character substitution, and lightly reformatted data. The use of shingles alone does not satisfy the one-wayness requirement. Rabin fingerprint is utilized to satisfy such requirement after shingling. In 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. *q(x)* is mod by a selected irreducible polynomial *p(x)*. The process shown in (1) maps a *k*-bit shingle into a *p f* -bit fingerprint *f* wherethe degree of *p(x)* is *p f* + 1.

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

From the detection perspective, a straight forward method
is for the DLD provider to raise an alert if any sensitive
fingerprint matches the fingerprints from the traffic.1 However, this approach has a privacy issue. If there is a data leak, there is a match between two fingerprints from sensitive data and network traffic. Then, the DLD provider learns the corresponding shingle, as it knows the content of the packet. Therefore, the central challenge is *to prevent the DLD provide* from learning the sensitive values even in data-leak scenarios, while allowing the provider to carry out the traffic inspection.

We propose an efficient technique to address this problem. The main idea is to relax the comparison criteria by strategically introducing matching instances on the DLD provider's side *without increasing false alarms for the data owner*.
Specifically,
i) the data owner perturbs the sensitive-data fingerprints before disclosing them to the DLD provider,and
ii) the DLD provider detects leaking by a range-based
comparison instead of the exact match.
The range used in the comparison is pre-defined by the data owner and correlates to the perturbation procedure.

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

Definition 2: Given a fuzzy length pd , and a collection of
fingerprints, the fuzzy set S f,pd of a fingerprint f is the set of fingerprints in the collection whose values differ from f by at most 2pd − 1.

In Definition 2, the size of the fuzzy set $|S_{f,pd}|$ is upper
bounded by 2^{pd} , but the actual size may be smaller due to the sparsity of the fingerprint space.

OPERATIONS IN OUR PROTOCOL

1) **PREPROCESS**: This operation is run by the owner on each piece of sensitive data. 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 $p_f + 1$) used in Rabin fingerprint. Each fingerprint is p_f –bit long and the fuzzy length is p_d . *M* is a bitmask, which is p_f-bit long and contains p_d `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 fingerprints of the piece of sensitive data.
c) The data owner transforms each fingerprint $f \in S$ into a fuzzy fingerprint f_* with randomized bits (specified by the mask *M*). The procedure is described as follows: for each $f \in S$, the data owner generates a random p_f-bit binary string f', mask out the bits not randomized by $f' = (\texttt{NOT}\ M)\ \texttt{AND}\ f$ (`1`'s in *M* indicate positions of bits not to randomize), and fuzzify *f* with $f_* = f\ \texttt{XOR}\ f'$. The overall computation is
described in (2).

$$f_* = ((\texttt{NOT}\ M)\ \texttt{AND}\ f')\ \texttt{XOR}\ f \ldots\ldots\ldots (2)$$

All fuzzy fingerprints are collected and form the output of this operation, the fuzzy fingerprint set, S_*.

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

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 $(q, p(x), p_d, M)$. The data owner keeps S for use in the subsequent POSTPROCESS operation.

3) **MONITOR**: This operation is run by the DLD provider. The DLD provider monitors the network traffic T from the data owner's organization. Each packet in T is collected and the payload of it is sent to the next operation as the network traffic (binary) string $\tilde{T}$. The payload of each packet is not the only choice to define $\tilde{T}$. A more sophisticated approach could identify TCP flows and extract contents in a TCP session as $\tilde{T}$. Contents of other protocols can also be retrieved if required by the detection metrics.

4) **DETECT**: This operation is run by the DLD provider on
each $\tilde{T}$ as follows.

a) The DLD provider first computes the set of Rabin fingerprints of traffic content $\tilde{T}$ based on the public parameters. The set is denoted as T.

b) The DLD provider tests whether each fingerprint $f\_ \in T$ is also in S_* using the fuzzy equivalence test (3).

$$E(f\_, f_*) = \texttt{NOT}\ (M\ \texttt{AND}\ (f\_\ \texttt{XOR}\ f_*)) \quad \ldots\ldots\ldots\ldots\ldots (3)$$

$E(f\_, f_*)$ is either `True` or `False`. $f\_$ `XOR` f_* gives the difference between $f\_$ and f_*.

M `AND` $(f\_$ `XOR` $f_*)$ filters the result leaving only the interesting bits (preserved bits with `1`'s in M). Because `XOR` yields `0` for equivalent bits, `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 fingerprint $f_* \in S_*$. (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 record in a set $\hat{T}$.

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

5) **REPORT**: If DETECTION on $\tilde{T}$ yields an alert, the DLD provider reports the set of detected candidate leak
instances $\hat{T}$ to the data owner.

6) **POSTPROCESS**: After receiving $\hat{T}$, the data owner test
every $f\_ \in \hat{T}$ to see whether it is in S.

In the protocol, because S_{f_*,p_d}, the fuzzy set of f_*, includes
the original fingerprint f, the true data leak can be detected
(i.e., true positive). Yet, due to the increased detection range, multiple values in S_{f_*,p_d} may trigger alerts. Because S_{f_*,p_d} is large for the given network flow, the DLD provider has a low probability of pinpointing the sensitive data.

The advantage of our method is that the additional matching instances introduced by fuzzy fingerprints protect the sensitive data from the DLD provider; yet they do not cause additional false alarms for the data owner, as the data owner can quickly distinguish true and false leak instances. Given the digest f of a piece of sensitive data, a large collection T of traffic fingerprints, and a positive integer $K \_ |T|$, the data owner can choose a fuzzy length p_d such that there are at least $K - 1$ other distinct digests in the fuzzy set of f, assuming that the shingles corresponding to these K digests are equally likely to be candidates for sensitive data and to appear in network traffic. A tight fuzzy length (i.e., the smallest p_d value satisfying the privacy requirement) is important for efficient POSTPROCESS operation. Due to the dynamic nature of network traffic, p_d needs to be estimated accordingly. There exists an obvious tradeoff between privacy and detection efficiency – large fuzzy set allows a fingerprint to hide among others and confuses the DLD provider, yet this indistinguishability results in more work in POSTPROCESS. We provide quantitative analysis on fuzzy fingerprint including empirical results on different sizes of fuzzy sets.[1]

III. CONCLUSION

In the present scenario advancement in information technology field made storage of the data including private as well as public in the digital form. Since these data may be sensitive and confidential in nature , there exists various threats that are focusing on break the confidentiality and privacy of these data. Among these threats data leakage is the severe and most common. There exists many methods that are used for detect data leakage through aid of third party software but protecting the confidential data is a major concern for the organizations and individuals, where the confidential data falls into fraudulent hands. In this paper we are proposing a privacy-preserving data-leak detection model that using the basics of fuzzy fingerprint method. The proposed method using special digests, the exposure

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

of the sensitive data is kept to a minimum during the detection.

REFERENCES

[1] Xiaokui Shu, Danfeng Yao, ―Privacy-Preserving Detection of Sensitive Data Exposure‖, IEEE TRANSA CTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 5, MAY 2015

[2]International Journal on Cybernetics & Informatics (IJCI) Vol. 5, No. 2, April 2016 DOI: 10.5121/ijci.2016.5207 61 FUZZY FINGERPRINT METHOD FOR DETECTION OF SENSITIVE DATA EXPOSURE Staicy Ulahannanl and Roshni Jose2

[3] Risk Based Security. (Feb. 2014). *Data Breach Quick-View: An Executive's Guide to 2013 Data Breach Trends*. [Online].Available: https://www.riskbasedsecurity.com/reports/2013-DataBreachQuickView.pdf, accessed Oct. 2014.

[4] Ponemon Institute. (May 2013). *2013 Cost of Data Breach Study: GlobalAnalysis*. [Online] Available: https://www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf, accessed Oct. 2014.

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

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