Fast Detection of Transformed Data Leaks framework on Mail Server

Deepa B¹, Prathisha R², Mohana M
UG Scholars, Department of Information and Technology,
Sri Vidya College of Engineering and Technology, Virudhunagar, Tamilnadu, India
Assistant Professor, Department of Computer Science and Engineering,
Sri Vidya College of Engineering and Technology, Virudhunagar, Tamilnadu, India
Deepabav1221@gmail.com¹, prathisharamesh@gmail.com², m.mohanamo@gmail.com

ABSTRACT

Statistics from security firms, research institutions and government organizations show that the numberof data-leak instances has grown rapidly in recent years. Among various data-leak cases, human mistakes are one of the main causes of data loss. According to a report from Risk Based Security (RBS) the number of leaked sensitive data records has increased dramatically during the last few years, i.e., from 412 million in 2012 to 822 million in 2013. Deliberately planned attacks, inadvertent leaks (forwarding confidential emails to unclassified email accounts) and human mistakes (assigning the wrong privilege) lead to most of the data-leak incidents.A recent Kaspersky Lab survey shows that accidental leak by staff is the leading cause for internal data leaks in corporates . The data-leak risks posed by accidents exceed the risks posed by vulnerable software. In order to minimize the exposure of sensitive data and documents, an organization needs to prevent cleartext sensitive data from appearing in the storage or communication.

Keywords:-sensitivedata,privacy,data detection, Levenshtein-distance

Introduction

In Networks ,Data leak detection content inspection, sampling, alignment, dynamic programming, parallelism. reports show that the number of leaked sensitive data records has grown 10 times in the last 4 years, and it reached a record high of 1.1 billion in 2014. A significant portion of the data leak incidents are due to human errors, for example, a lost or stolen laptop containing unencrypted sensitive files, or transmitting sensitive data without using endto-end encryption such as PGP. A recent Kaspersky Lab survey shows that accidental leak by staff is the leading cause for internal data leaks in corporates . The data-leak risks posed by accidents exceed the risks posed by vulnerable software. In order to minimize the exposure of sensitive data and documents, an organization needs to prevent cleartext sensitive data from appearing in the storage or communication.

A screening tool can be deployed to scan computer file systems, server storage, and inspect outbound network traffic. The tool searches for the occurrences of plaintext sensitive data in the content of files or network traffic. It alerts users and administrators of the identified data exposure vulnerabilities. For example, an organization's mail server can inspect the content of outbound email messages searching for sensitive data appearing in unencrypted messages. Data leak detection differs from the anti-virus (AV) scanning (e.g., scanning file systems for malware signatures) or the network intrusion detection systems (NIDS) (e.g., scanning traffic payload for malicious patterns) . AV and NIDS typically employ automata-based string matching (e.g., Aho-Corasick , Boyer-Moore , which match static or regular patterns.

Statistics from security firms, research institutions and government organizations show that the number of data-leak instances has grown rapidly in recent years. Among various data-leak cases, human mistakes are one of the main causes of data loss. According to a report from Risk Based Security (RBS) the number of leaked Sensitive data records has increased dramatically during the last few years, i.e., from 412 million in 2012 to 822 million in 2013. Deliberately planned attacks, inadvertent leaks (forwarding confidential emails to unclassified email accounts) and human mistakes (assigning the wrong privilege) lead to most of the data-leak incidents. Detecting and preventing data leaks requires a set of complementary solutions, which may include data-leak detection data confinement stealthy malware detection and policy enforcement. Network data-leak detection (DLD) typically performs deep packet inspection (DPI) and searches for any occurrences of Sensitive data patterns.

OBJECTIVE

The main aim of this project is to provide security against Sensitive data, using data leak detection technique on the transformed data. Detecting and preventing data leaks requires a set of complementary solutions, which may include data-leak detection data confinement stealthy malware detection and policy enforcement. Network data-leak detection (DLD) performs and searches for any occurrences of sensitive data patterns.

LITERATURE SURVEY

X Shu, D Yao, E Bertin-"Privacy-preserving detection of sensitive data exposure", Statistics from security firms, research institutions and government organizations show that the number of data-leak instances have grown rapidly in recent years. Among various data-leak cases, human mistakes are one of the main causes of data loss. There exist solutions detecting inadvertent sensitive data leaks caused by human mistakes and to provide alerts for organizations. A common approach is to screen content in storage and transmission for exposed sensitive information. Such an approach usually requires the detection operation to be conducted in secrecy. This secrecy requirement is challenging to satisfy in practice, as detection servers may be compromised or outsourced. In this paper, we present a privacy-preserving data-leak detection (DLD) solution to solve the issue where a special set of sensitive data digests is used in detection. The advantage of our method is that it enables the data owner to safely delegate the detection operation to a semihonest provider without revealing the sensitive data to the provider. We describe how Internet service providers can offer their customers DLD as an add-on service with strong privacy guarantees.

Xiaokui Shu, Jing Zhang, Danfeng (Daphne) Yao, Wu-Chun Feng- "Rapid Screening of
Kevin Border, Inc,Ann Arbor-" Quantifying Information Leaks in Outbound Web Traffic", As the Internet grows and network bandwidth continues to increase, administrators are faced with the task of keeping confidential information from leaving their networks. Today's network traffic is so voluminous that manual inspection would be unreasonably expensive. In response, researchers have created data loss prevention systems that check outgoing traffic for known confidential information. These systems stop naïve adversaries from leaking data, but are fundamentally unable to identify encrypted or obfuscated information leaks. What remains is a high-capacity pipe for tunneling data to the Internet. We take advantage of the insight that most network traffic is repeated or determined by external information, such as protocol specifications or messages sent by a server. By filtering this data, we can isolate and quantify true information flowing from a computer. In this paper, we present measurement algorithms for the Hypertext Transfer Protocol (HTTP), the main protocol for web browsing. When applied to real web browsing traffic, the algorithms were able to discount 98.5% of measured bytes and effectively isolate information leaks.

Somesh Jha- "Towards Practical Privacy for Genomic Computation", Many basic tasks in computational biology involve operations on individual DNA and protein sequences. These sequences, even when anonymized, are vulnerable to re-identification attacks and may reveal highly sensitive information about individuals. We present a relatively efficient, privacy-preserving implementation of fundamental genomic computations such as calculating the edit distance and Smith- Waterman similarity scores between two sequences.

Our techniques are cryptographically secure and significantly more practical than previous solutions. We evaluate our prototype implementation on sequences from the Pfam database of protein families, and demonstrate that its performance is adequate for solving real-world sequence-alignment and related problems in a privacypreserving manner. Furthermore, our techniques have applications beyond computational biology. They can be used to obtain efficient, privacy-preserving implementations for many dynamic programming algorithms over distributed datasets.

Sailesh Kumar, Balakrishnan Chandrasekaran,Jonathan Turner-"Curing RegularExpressions Matching Algorithms from Insomnia, Amnesia, and Acalculia",The importance of network security has grown tremendously and a collection of devices have been introduced, which can improve the security of a network. Network intrusion detection systems (NIDS) are among the most widely deployed such system; popular NIDS use a collection of signatures of known security threats and viruses, which are used to scan each packet's payload. Today, signatures are often specified as regular expressions; thus the core of the NIDS comprises of a regular expressions parser; such parsers are traditionally implemented as finite automata. Deterministic Finite Automata (DFA) are fast, therefore they are often desirable at high network link rates. DFA for the signatures, which are used in the current security devices, however require prohibitive amounts of memory, which limits their practical use. In this paper, we argue that the traditional DFA based NIDS has three main limitations: first they fail to exploit the fact that normal data streams rarely match any virus signature; second, DFAs are extremely inefficient in following multiple partially matching signatures and explodes in size, and third, finite automaton are incapable of efficiently keeping track of counts. We propose mechanisms to solve each of these drawbacks and demonstrate that our solutions can implement a NIDS much more securely and economically, and at the same time substantially improve the packet throughput

PROPOSED SYSTEM

In our proposed system we propose a data-leak detection solution which can be outsourced from organization, we design and implement Lucerne search engine framework Levenshtein-distance technique to avoid data leak and also provide privacy preserving to Sensitive data. Two most important players in our proposed model is

- **Data Owner** owns the Sensitive data and authorizes the DLD provider to inspect the network traffic from the organizational networks for anomalies, namely inadvertent data leak.
- **Mail Server - DLD provider** inspects the network traffic for potential data leaks. We focus on detecting inadvertent data leaks, and we assume the content in file system or network traffic 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. Authority has the threshold for every categorized position of users
- In our security model, we assume that the analysis system is secure and trustworthy. Privacy-preserved data-leak detection can be achieved by leveraging special protocols and computation steps. It is another functionality of a detection system
- We implement the web service to maintain the users and Sensitive content instead of data bases because of static implementation and rough data handling. Even the Sensitive data storage have to preserved from threatens in existing system. For that purpose we used to maintain the Sensitive data in cloud.

SYSTEM ARCHITECTURE:

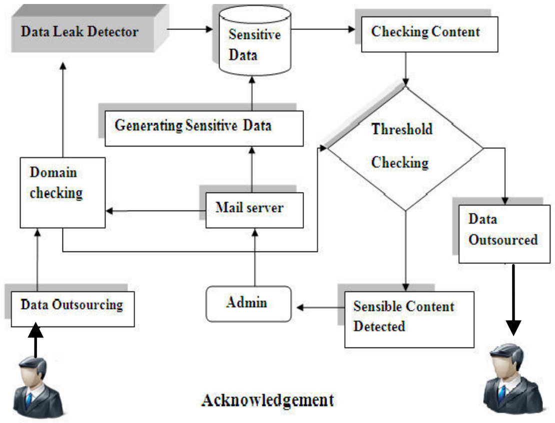

Fig 1 system architecture

Modules:

- ➢ Create user login
- ➢ Build data leakage detection framework.
- ➢ Content Outsourcing with DLD Checker.
- ➢ Sensitive Data Detection.

Create User Login:

In this module user's register in mail server with their name, authorized job position and their authorized e-mail domain. And the users can transfer their file using without any restriction of sensible content checking.

There is no content checking and domain filtering on their transformed sensible data. Sensible content is outsourcing from one organization to another organization performed by user. The content can be of any file (text, document). Outsourcing will not reach DLD and directly reach its destination or organization. Here outsourcing mechanism of transferred data is offending over the protocol.

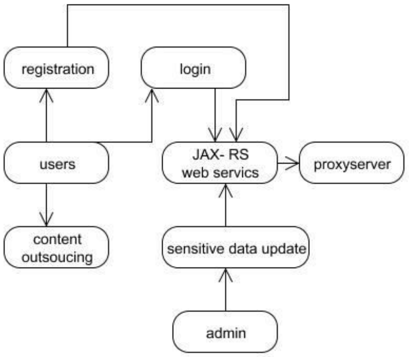

Fig 2 flow of create user login

Build data leakage detection framework:

In this module mail server data owner generates a sensitive data and stored in the cloud and create the directory for lucene search framework and other data leakage detectors. Data owner's cloud contains much sensitive information about their authorized customer's details, information technology source, and database and server details. This sensitive information is maintained by Data Leak Detector. Using this DLD referenced directory perform data leak detection mechanism.

The DLD consist of lucene search engine framework, levenshtein distance algorithm and our own shuffled checking algorithm. The DLD directly configured with cloud and can refer every data transformation outsourcing from authorized user transformation.

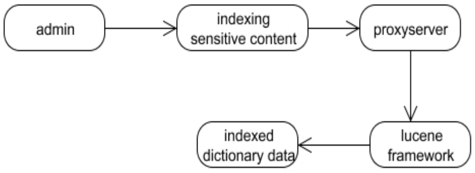

Fig 3 flow of data leakage detection framework

Content Outsourcing with DLD Checker:

DLD is the one will check all the outsourcing content before it transmit to the other organization. All the outsourced contents are check with sensitive data. All the sensitive data are maintaining in index file. Using this index file DLD identify the sensitive data concurrently with domain filtering and threshold assigning based on their email domain. DLD will check every line of the sending data with the sensitive file. DLD will not allow any sensitive data will leak to any of the other organization.

In proxy mail server the every occurrence of transformed contents are filter by users email domain. All users details are retrieved from the cloud using their email. Then threshold assigned for the users based on their authorized job position and the transferred content has been tested by lucene framework search engine, levenshtein distance checking and shuffling algorithm.

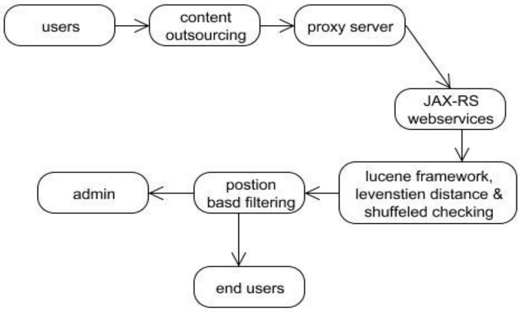

Fig 4 flow of Content Outsourcing with DLD Checker

Sensitive Data Detection:

Once the DLD framework checks the outsourced content, if any data leak is identified means DLD will detect the sensitive data. Here DLD will check not only the sensitive data and also it will check some access condition. Every data owner maintain common access condition every file. For example, all the contents are encrypted before they outsourced. If DLD identified any sensitive information outsourcing means they will detect the sensible content in between of the file outsourcing.

For the purpose of false alert, we maintain threshold of every domain and users position. If the sensible content percentage of transferred file exceeds the threshold percentage which trigger alert mail to Admin of the proxy mail server. Alert mail consists of entire details about the users even what are the sensible contents are pings from the transferred content by the DLD framework.

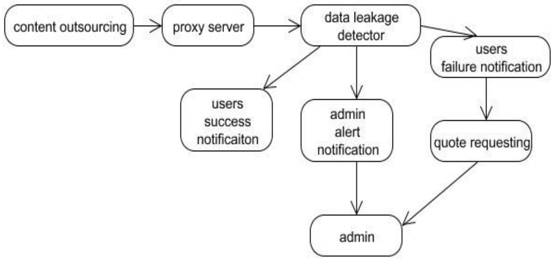

Fig 5 Sensitive Data Detection

Conclusion

Fast detection of data-leakage framework to avoid sensitive data exposure and also provide privacy-preserving to sensitive data. Lucerne search framework to detect the sensible data easily using indexing technique. Levenshtein distance algorithm to detect the shuffling of transferred mail content.Tos implement the own logics for detect sampling of n-grams in transferred mail content appropriately.We implement threshold rate based on assigning and checking domains based user filtering technique.

REFERENCES

[1] 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.

[2] X. Shu, J. Zhang, D. Yao, and W.-C. Feng, "Rapid screening of transformed data leaks with efficient algorithms and parallel computing," in *Proc. 5th ACM Conf. Data Appl. Secur. Privacy (CODASPY)*, San Antonio, TX, USA, Mar. 2015, pp. 147–149.

[3] (Feb. 2015). *Data Breach QuickView: 2014 Data Breach Trends*. [Online]. Available: https://www.riskbasedsecurity.com/reports/2014-YEDataBreachQuickView.pdf, accessed Feb. 2015.

[4] S. Kumar, B. Chandrasekaran, J. Turner, and G. Varghese, "Curing regular expressions matching algorithms from insomnia, amnesia, and acalculia," in *Proc. 3rd ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, 2007, pp. 155–164.

[5] L. De Carli, R. Sommer, and S. Jha, "Beyond pattern matching: A concurrency model for stateful deep packet inspection," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 1378–1390.

[6] A. V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," *Commun. ACM*, vol. 18, no. 6, pp. 333–340, Jun. 1975.

[7] R. S. Boyer and J. S. Moore, "A fast string searching algorithm," *Commun. ACM*, vol. 20, no. 10, pp. 762–772, Oct. 1977.