International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 11, November 2014 **ISSN 2319 - 4847**

Data alteration detection: A PC to USB Pen drive Perspective

Imran Shoukat Kazi[1], Zakir Mujeeb Shaikh[2]

[1]Department of Computer Science and Engineering,
NK Orchid College of Engineering & Technology Solapur, Maharashtra, India

[2]Department of Computer Science and Engineering,
NK Orchid College of Engineering & Technology Solapur, Maharashtra, India

ABSTRACT

Protecting sensitive information from unauthorized user is a major concern of every organization. As an organization's employees need to access such information in order to carry out their daily work. Some time the critical data can carry by employee to work from home. There are several ways to carry the data i.e. using USB Pen drive/HDD, CD/DVD, E-mail or using VPN etc. The security of data is more important when the employee carry the data to work from home. There are few records which not be allowed to alter that record should be consider as reference e.g. finance data. If such record can be modified then origination significantly damage. Data alteration detection is an essential and challenging task. Their are several approaches for protecting data alteration like Fingerprinting, USB block etc. In Fingerprinting signatures of known confidential content are extracted and matched with outgoing content in order to detect leakage of sensitive content. In this paper we propose a method for detecting the data alteration based on SHA1 algorithm.

Keywords: Secure Hashing Algorithm(SHA), Data Leakage Detection(DLD), Data Leakage Protection(DLP) ,Universal Serial Bus (USB), Hard Disk Drive(HDD), Virtual Private Network.

1. INTRODUCTION

Providing security to the data is important because sometimes sensitive data must be given to employee for analysis from home. For example, a company may give business data which is very confidential then there is high possibility of leakage of data [1]. Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious. In this paper, we study unobtrusive techniques for preventing data leakage of a set of objects or records.

2. LITERATURE SURVEY

The guilt detection approach is related to the data provenance problem [3],[5]: tracing the leakage of records implies essentially the detection of the guilty agents. And assume some prior knowledge on the way a data view is created out of data sources. There are lots of other works on mechanisms that allow only authorized users to access sensitive data through access control policies [4]. Such approaches prevent in some sense data leakage by sharing information only with trusted parties. However, these policies are restrictive and may make it impossible to satisfy agent's requests.

3. EXISTING SYSTEM

In existing system, we consider applications where the original sensitive data cannot be made less sensitive. However, in some cases it is important not to alter the original distributor's data. Traditionally, leakage detection is handled by giving a unique code and it is embedded within distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified data. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious.

4. PROPOSED SYSTEM

Our goal is to detect data leakage, when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data exclusive for USB pen drive. Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. We propose to develop unobtrusive techniques for detecting leakage of a set of objects or records. In this section, we propose to develop a model for

assessing the data alteration. We also present algorithms for distributing records to agents, in a way that improves our chances of identifying a leaker.

4.1 Problem Setup and Notation

A distributor owns a set of valuable files say F= {f1,f2,f3 ….fn} in his personal /company pc. The distributor wants to share same file with set of agents A1,A2,A3….An but wish the file will not be altered .

4.2 Algorithms

We require flowing algorithms :

i. Distributor to agent

Transferring Data from PC to Pen drive

Step 1. Click the File which you want to transfer

Step 2. Send the file to the destination.

Step 3. Calculate the hash value of each file before Copied to destination.

Step 4. Get the details of destination Pen drive

Step 5. Open the database

Step 6. Store the pen drive information along with hash values of file and time stamp.

Step 7. Log will me mentioned for each I/O Operation

Step 8. End

ii. Agent to Distributor

Transferring Data from Pen drive to PC

Step 1. Click the File which you want to transfer

Step 2. Send the file to the destination.

Step 3. Calculate the hash value of each file.

Step 4. Get the details of destination Pen drive

Step 5. Open the database

Step 6. Compare the hash values and pen drive details with existing database value

Step 7. If data is same No alteration made in USB

Step 8. If Yes Pen drives data is modified.

Step 9. Log will me mentioned for each I/O Operation

Step 10. End

iii. SHA1

With every record which is going to transfer from Personal computer to pen drive the hash value of each file is computed. Using SHA1 algorithm. SHA1 stand for “Secure Hashing Algorithm”. SHA1 is currently the most widely used in a wide variety of applications, including TLS, SSL, SSH and PGP. The SHA1 implemented using C#.

```
private static string GetSHA1(string text)
{
    UnicodeEncoding UE = new UnicodeEncoding();
    byte[] hashValue;
    byte[] message = UE.GetBytes(text);

    SHA1Managed hashString = new SHA1Managed();
    string hex = "";
     hashValue = hashString.ComputeHash(message);
    foreach (byte x in hashValue)
    {
       hex += String.Format("{0:x2}", x);
    }
    return hex;
        }
```

The benefit of hash will and kept in one database for further reference. The database should be made according to PNID, hash code and file name. When the agent put the pen drive the value of changed file will be displayed based on hash value which is already stored in database.

e.g.
The distributor want to transfer the source file name test.txt to destination pen drive i.e.:

According to the Distributor to agent algorithm
File name: test.txt
Contents: this is test file.
SHA1 Value: 731E5ED592C15DE3461AF45D96456AC390A33E09
PNID :
JDWZLGE5&0
Time Stamp : 2014-08-03 3:00 PM
If agent modified the content of the file say:
Contents: this is test file**s**.
PNID:
JDWZLGE5&0
SHA1 Value: 26B10125AA21728C2B5C9620B3626CE8888234F9
Apply string comparison

```
private static bool isStringEqual(string a, string b)
{
    if (a.Equals(b))
            return true;
    return false;
}
```

A small change made by agent in the content will result the different hash value and if we apply the string comparison method we will get false i.e. two hash values are not matching. And hence we concluded that file is altered.

5. CONCLUSION

From this study and implementation of the algorithm, we conclude that the data alteration will damage the origination significantly. At the same time data leakage detection industry is very heterogeneous as it evolved out of ripe product lines of leading IT security vendors. A broad arsenal of enabling technologies such as firewalls, encryption, access control, identity management, machine learning content/context-based detectors and others have already been incorporated to offer protection against various facets of the data leakage threat. The competitive benefits of developing a "one-stop-shop", silver bullet data leakage detection suite is mainly in facilitating effective orchestration of the aforementioned enabling technologies to provide the highest degree of protection by ensuring an optimal fit of specific data leakage detection technologies with the "threat landscape" they operate in. This landscape is characterized by types of leakage channels, data states, users, and IT platforms. I also consider, the option of adding “fake” objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents. In a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that Ire leaked, then the distributor can be more confident that agent was guilty.

6. ACKNOWLEDGEMENT

For all the efforts behind the paper work, we first & foremost would like to express our sincere appreciation to the staff of Dept. of Computer Sci.& Engg. NK Orchid College of Engineering and Technology Solapur, Maharashtra, India, for their extended help & suggestions at every stage of this paper. It is with a great sense of gratitude that we acknowledge the support, time to time suggestions. Finally, we pay sincere thanks to all those, who indirectly and directly helped me towards the successful completion of the paper.

REFERENCES

[1] Panagiotis Papadimitriou, Hector Garcia-Molina , IEEE Paper “Data Leakage Detection”,2011.
[2] Panagiotis Papadimitriou, Hector Garcia-Molina , IEEE Paper “Data Leakage Detection”,2010.
[3] P. Buneman, S. Khanna, and W.C. Tan, “Why and Where: A Characterization of Data Provenance,” Proc. Eighth Int’l Conf. Database Theory (ICDT ’01), J.V. den Bussche and V. Vianu, eds., pp. 316-330, Jan. 2001.

[4] P. Bonatti, S.D.C. di Vimercati, and P. Samarati, "An Algebra for Composing Access Control Policies," ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.
[5] P. Buneman and W.-C. Tan, "Provenance in Databases," Proc. ACM SIGMOD, pp. 1171-1173, 2007.

AUTHOR

Imran Shoukat Kazi received the B.E. Computer Science and Engineering from Solapur University,Solapur Maharashtra, India. and pursing Master of Engineering in Computer Science and Engineering from Nagesh Karajagi Orchid College of Engineering and Technology, Solapur, Solapur University Solapur, Maharashtra India.

Zakir Mujeeb Shaikh received the B.E. Computer Science and Engineering from Dr. Babasaheb Ambedkar University, Aurangabad Maharashtra, India and Master of Engineering in Computer Engineering from Bharati Vidyapeeth Pune. His area of research is Software engineering, Database, Programming and Image processing. The author is having more the fourteen years of experience.