Clipboard Monitoring for Data Loss Prevention

M.Asha[1], Dr.J.Sreerambabu[2], M.Mohammed Riyaz[3]
[1]Assistant Professor, [2] Head of the Department, [3]Assistant Professor
Master of Computer Applications Department
Thanthai Periyar Government. Institute of Technology, Vellore-2,India

Abstract

This paper is an attempt to survey and study Data leakage/loss prevention (DLP). Data leakage causes negative impact on companies. The traditional security approaches, such as firewalls, can't protect data from leakage. DLP systems are solutions that protect sensitive data from being in non-trusted hands. Data loss in an organization is a big risk for the company and also for the clients. Maintaining the confidential data and preventing it. It is proposed that a software system "Clip board monitoring" may be developed to nurture the needs of an IT Company so that they can prevent their confidential data from outsiders (unauthorized users), malicious insiders and others. This software is also developed to nurture the needs of the organization so that they can prevent data leak through external devices, network and internet.

Keywords: *Data Loss Prevention, confidential data, policy pushing.*

I. INTRODUCTION

An application is built in the request of the IT organization to prevent their data (Confidential data) from outsiders(unauthorized users), malicious insiders and by other means. The organization requested to built to prevent data leak through external devices, network and internet.

This software is based on client-server communication. The software push policies to the client systems from the server system. The activities of client systems are monitored from the server system.

The concentration will be on highlighting the events starting from choosing the type of Data Loss Prevention tool. Each tool has its own rules for preventing data leakage.

The major component of the "Data Loss Prevention" is End Point Protection. The End Point Protection tool prevents the data leakage through external devices. This tool monitors the clipboard, foreground window and file focus. It validates the files when user performs some action. It validates the file to find the presence of confidential data. It blocks the data from applying the user action, if it senses some confidential data. Otherwise it allows the user to perform the action.

The confidential data are specified in Policies (Rules) and the DLP tool also allows the user to create his/her own rules apart from the default policies. These pre-defined and user-defined policies are pushed from server to client. The pre-defined policies are Regular expression for confidential data like Credit card, Bank account number and etc,. The user-defined policies are Keyword matching (for contractual document) and user defined Regular expressions (User creates his/her own pattern).

The organization does want to prevent their data, but they also need to work with it. So DLP tool give some option as MODE (Blocking, Monitoring and Limited Monitoring). The users put in Blocking mode are monitored and they are not allowed to copy file (or perform some action in the file) containing confidential data. The users in Monitoring mode are also monitored but they are allowed to copy file (or perform some action in the file) containing confidential data. The users in Limited Monitoring mode are monitored and they can copy or perform some action with confidential data but they are restricted with some limits. If they cross the limit of data, they are blocked from performing the action.

II. EXISTING SYSTEM ARCHITECTURE

The Data Loss Prevention is about preventing sensitive data from unauthorized users. The data security is about protecting the confidential data or information within the company. Generally the confidential data can be protected with password, personal accounts, data encryption etc,. Though the data are protected but still there are possibilities to steal the data from an organization.

The existing systems protect data by blocking entire data transfer from any system.

It does not provide any policy to check confidential data. It makes difficult for any data transfer within an organization.

The existing system does not have separate policies to validate a confidential data. It makes easier for a malicious insider to transfer confidential data outside the organization.

The existing system does not monitor the client machine activities to analyze the type of data being transferred. Hence the organization cannot find the malicious insider easily.

Need for New System

In the proposed system, monitors and blocks only sensitive data. We need to create individual policy for each category. We have to monitor each user connected to the network in an organization.

III. PROPOSED SYSTEM ARCHITECTURE

Data Loss Prevention is a recent type of security technology that works toward securing sensitive data in an automated and non-intrusive fashion. Through policies a DLP system automatically makes sure no sensitive data is stored, sent or accessed where it shouldn't be, while still allowing users to use the tools and services they choose and need to fulfil their tasks. Unlike traditional white and blacklisting, the DLP only blocks the actions where sensitive data is involved, e.g. sending e-mails is perfectly acceptable, but not if they contain sensitive data. DLP can also be set to handle different levels of sensitivity and document access control. DLP systems keep user from deliberately or inadvertently sending out sensitive material without authorization. DLP software products use business rules to classify and protect confidential and critical information so that unauthorized end users cannot accidentally or maliciously share data whose disclosure could put the organization at risk.

The proposed systems protect data by blocking only confidential data transfer from any system. It process under policy to validate and transmit a confidential data.The proposed system creates several policies to validate and prevent a confidential data. It makes easier to identify a malicious insider from transmitting confidential data outside the organization.The proposed system monitors all the activities on the client machine to analyze the type of data being transferred.

The proposed system monitors the transmitting any data through E-Mail. Transmitting any data outside the organization is not easy. It validates the attachment of any file containing confidential data through personal mail. If sensitive data occurs it blocks the mail from sending and generates an alert message to the Admin.

Advantage of Proposed System

- DLP monitors and blocks only sensitive data.
- Individual policy for each category.
- Monitor each user connected to the network

IV. CLIPBOARD MONITORING

The clipboard is a software facility used for short-term data storage and/or data transfer between documents or applications, via copy and paste operations. It is most commonly a part of a GUI environment and is usually implemented as an anonymous, temporary data buffer, sometimes called the paste buffer, that can be accessed from most or all programs within the environment via defined programming interfaces. A typical application accesses clipboard functionality by mapping user input (keybindings, menu selections, etc.) to these interfaces.

Clipboard managers are applications that enable user to manipulate clipboard.

If you use CTRL+C, some data or files are copied to a system clipboard and stored in system memory and when you use CTRL+V, the data is copied back to wherever you paste it. The Clipboard class provides functionality to place and retrieve data from the system clipboard.

In present time internet is backbone of our society. From a child to old man all are somehow connected to internet, because internet is like ocean of information. Everybody needs internet, as it is growing with exponential rate and web applications are also growing. On the other side malicious author are also making internet users as their prime target. People used to perform copy/cut and paste operation much frequently then typing anything making the use of clipboard. The clipboard is the common operating system component which enables application to transfer data among other. An application places data into the clipboard with the cut or copy operations and other applications can retrieve the data from the clipboard with the paste operation. So user's habit to cut/ copy paste provides malicious application and malware a loophole in security at client based system. So security of important data is increasingly demanded, data transmitted between independent programs through the clipboard needs to be protected. There can be many attack or applications which can register themselves as clipboard event viewer and observe the clipboard content and can capture to modify it. In attack name Hitchbot they deliver malicious content by getting the clipboard content and modify it with similar looking malicious content. So security to clipboard content is issue of concern and need to enhance the functionality of operating system by which we can protect our clipboard data from being got watched.

A. Methodology

There are many different ways possible to extract data from clipboard, one of which is described below using which the data can be extracted and identified. The below fig describes steps required for extracting data. First, functions that used to access clipboard are identified. Then the clipboard data is accessed using those functions and the type of data is identified depending on its properties. The data retrieved is classified in different formats and if the required format is found in the clipboard then it is tagged. Finally the identified data is extracted and used depending on applications.

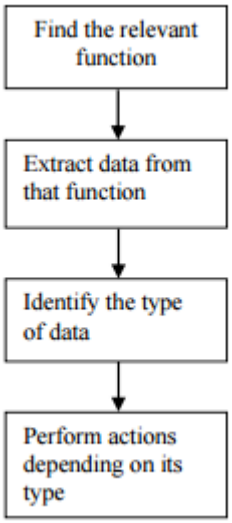

Fig 1: Process of Extracting Data From Clipboard

B. Clipboard Classes and Methods

While there is a large amount of documentation on how to use and access the Windows clipboard via application program interfaces (APIs), there is less documentation on the methods used other than APIs. Extending classes like Clipboard we can access the data stored in the clipboard using its constructions and different methods. The getSystemClipboard is called to retrieve the pointer to the clipboard data. This provides handle to the clipboard data. The data can then be used to analyse type and its properties. The Toolkit class is the main class which is use to get the system clipboard. A DataFlavor provides Meta information about data. Data Flavor is typically used to access data on the clipboard or during drag and drop operation. FlavorListeners may be registered on an instance of the Clipboard class to be notified about the changes to the set of DataFlavors available on this Clipboard. There are different data flavors that provide the type of data that is stored in the clipboard. Those flavors are classified depending on file formats. IsDataFlavorAvailable method is used to check whether the data flavor of the data in the clipboard is in the list of valid data flavors. Manipulating the clipboard data is quiet easy but it will be not useful in applications where the memory forensics is required. So care needs to be taken while handling pointer of the clipboard which is directly accessed.

C. Implementation

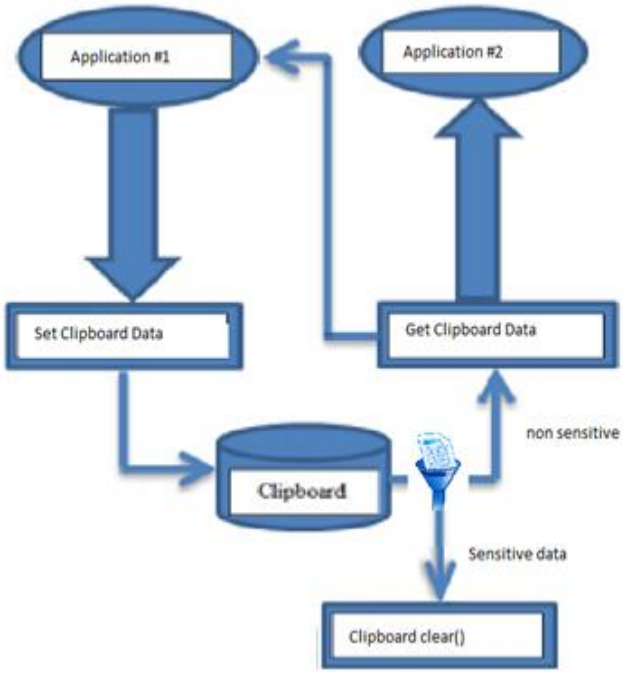

Fig 2: Working Mechanism of Clipboard

The user tries to copy data from any application. When the data is copied, the method setclipboarddata() is called and the data is stored in clipboard. In our DLP Endpoint Protection When the user tries to copy any data.

The data is read from clipboard to validate whether the data in clipboard is sensitive or not .
If the clipboard contains sensitive data it calls the clipboard clear() method to clear the sensitive data.

If the clipboard contains normal data it calls the getclipboarddata() method to allow copy.

Using above classes and methods the data can be obtained from the clipboard. The below algorithm can be used as the basic layout for the working of extraction.

```
Clipboard c;

 c=getSystemClipboard();

if(DataFlavor is Available) getdata(DataFlavor);
```

It is possible that the user closed the application after copying data to the clipboard. Recall that the clipboard bridges user space and kernel space. While

each process has a local copy of the clipboard once it has accessed the clipboard functions, the kernel also has the clipboard. Therefore, until overwritten, clipboard data for a closed process is still available in the clipboard. It is also possible to change the contents of the clipboard using setContents method. But the owner on the current control is shifted to the application which modifies the content of the clipboard data. For e.g.

Clipboard c;

StringSelectioncnt=newStringSelection("Data");

c.setContents(cnt);

As shown above the current data flavor is been replaced by the String flavor as the strings "Data" is been placed in the clipboard replacing the older data. Clipboard class is much useful for altering data in the clipboard without letting the user realise about it. It can be also used for the applications which are based on monitoring the clipboard and restricting the contents to be copied

D. How to: Retrieve Data from the Clipboard

The Clipboard class provides methods that you can use to interact with the Windows operating system Clipboard feature. Many applications use the Clipboard as a temporary repository for data. For example, word processors use the Clipboard during cut-and-paste operations. The Clipboard is also useful for transferring information from one application to another.

Some applications store data on the Clipboard in multiple formats to increase the number of other applications that can potentially use the data. A Clipboard format is a string that identifies the format. An application that uses the identified format can retrieve the associated data on the Clipboard. The DataFormats class provides predefined format names for your use. You can also use your own format names or use an object's type as its format. For information about adding data to the Clipboard, see How to: Add Data to the Clipboard.

To determine whether the Clipboard contains data in a particular format, use one of the ContainsFormat methods or the GetData method. To retrieve data from the Clipboard, use one of the GetFormat methods or the GetData method. These methods are new in .NET Framework 2.0.

To access data from the Clipboard by using versions earlier than .NET Framework 2.0, use the GetDataObject method and call the methods of the returned IDataObject. To determine whether a particular format is available in the returned object, for example, call the GetDataPresent method.

E. To Retrieve Data from the Clipboard in a Single, Common Format

Use he GetAudioStream, GetFileDropList, GetImage, or GetText method. Optionally, use the corresponding Contains Format methods first to determine whether data is available in a particular format. These methods are available only in .NET Framework 2.0.

F. To Retrieve Data from the Clipboard in a Custom Format

Use the GetData method with a custom format name. This method is available only in .NET Framework 2.0. You can also use predefined format names with the SetDatamethod. For more information, see DataFormats.

G. To Retrieve Data from the Clipboard in Multiple Formats

Use the GetDataObject method. You must use this method to retrieve data from the Clipboard on versions earlier than .NET Framework 2.0.

H. Drag and Drop, and the Clipboard

The clipboard is an object where programs can save and restore data. A program can save data in multiple formats and retrieve it later, or another program might retrieve the data. Windows, rather than Visual Basic, provides the clipboard, so it is available to every application running on the system, and any program can save or fetch data from the clipboard.

The clipboard can store remarkably complex data types. For example, an application can store a representation of a complete object in the clipboard for use by other applications that know how to use that kind of object.

Drag - and - drop support enables the user to drag information from one control to another. The controls may be in the same application or in different applications. For example, your program could let the user drag items from one list to another, or it could let the user drag files from Windows Explorer into a file list inside your program.

A drag occurs in three main steps. First, a drag source control starts the drag, usually when the user presses the mouse down on the control. The control starts the drag, indicating the data that it wants to drag and the type of drag operations it wants to perform (such as Copy, Link, or Move

When the user drags over a control, that control is a possible drop target. The control examines the kind of data being dragged and the type of drag operation requested (such as Copy, Link, or Move). The drop target then decides whether it will allow the drop and what type of feedback it should give to the user. For example, if the user drags a

picture over a label control, the label might refuse the drop and display a no drop icon (a circle with a line through it). If the user drags the picture over a PictureBox that the program is using to display images, it might display a drop link icon (a box with a curved arrow in it).

Finally, when the user releases the mouse, the current drop target receives the data and does whatever is appropriate. For example, if the drop target is a TextBox control and the data is a string, the TextBox control might display the string. If the same TextBox control receives a file name, it might read the file and display its contents.

The following sections describe drag - and - drop events in more detail and give several examples of common drag - and - drop tasks.

The section**"** Using the Clipboard **"**near the end of the chapter explains how to use the clipboard. Using it is very similar to using drag and drop, although it doesn't require as much user feedback, so it is considerably simpler.

V. DRAG - AND - DROP EVENTS

The drag source control starts a drag operation by calling its DoDragDrop method. It passes this method the data to be dragged and the type of drag operation that the control wants to perform. The drag type can be Copy, Link, or Move.

If you are dragging to other general applications, the data should be a standard data type such as a String or Bitmap so that the other application can understand it. If you are dragging data within a single application or between two applications that you have written, you can drag any type of data. This won't necessarily work with general objects and arbitrary applications. For example, WordPad doesn't know what an Employee object is, so you can't drop an Employee on it.

As the user drags the data around the screen, Visual Basic sends events to the controls it moves over. Those controls can indicate whether they will accept the data and how they can accept it. For example, a control might indicate that it will allow a Copy, but not a Move. The following table describes the events that a drop target receives as data is dragged over it.

A drop target with simple needs can specify the drop actions it will allow in its DragEnter event handler and not provide a DragOver event handler. It knows whether it will allow a drop based solely on the type of item being dropped. For example, a graphical application might allow the user to drop a bitmap on it, but not a string.

A more complex target that must track such items as the keyboard state, mouse position, and mouse button state can provide a DragOver event handler and skip the DragEnter event handler. For example, a circuit design application might check the drags position over its drawing surface, and highlight the location where the dragged item would be positioned. As the user moves the object around, the DragOver event would continue to fire so the program could update the drop highlighting.

After the drag and drop finishes, the drag source's DoDragDrop method returns the last type of action that was displayed when the user dropped the data. That lets the drag source know what the drop target expects the source to do with the data. For example, if the drop target accepted a Move, the drag source should remove the data from its control. If the drop target accepted a Copy, the drag source should not remove the data from its control.

The following table describes the two events that the drag source control receives to help it control the drop.

A. Learning Data Types Available

When the user drags data over a drop target, the target's DragEnter event handler decides which kinds of drop to allow. The event handler can use the e.GetDataPresent method to see whether the data is available in a particular format and then decide whether it should allow a drop accordingly.

GetDataPresent takes as a parameter a string giving the desired data type. An optional second parameter indicates whether the program will accept another format if the system can derive it from the original format. For example, the system can convert Text data into System.String data so you can decide whether to allow the system to make this conversion.

The DataFormats class provides standardized string values specifying various data types. For example, DataFormats.Text returns the string Text representing the text data type.

If you use a DataFormats value, you don't need to worry about misspelling one of these formats. Some of the most commonly used DataFormats include Bitmap, Html, StringFormat, and Text.

VI. CONCLUSION

In this Paper, attempt was made to prevent data leakage and unauthorized access to confidential data. we validated the documents that are copied from system to any of the external devices. And hence, blocked the confidential data transferred to external devices. And also prevent the access of server database, if it is confidential data. Hence, we have achieved Data Loss Prevention objectives.

REFERENCES

[1] Christian Nagel, Bill Evjen, Jay Glynn, "Professional C# 2008" Wrox Publishers [2008].

[2] Mathew MacDonald, "The Complete Reference ASP.Net" Tata McGrawHill Edition [2002].

[3] Ron Schmelzer, Chad Darby, Sam Hunting, "XML and Web Services" Pearson Education[2008].

[4] Dave Mercer, "ASP 3.0 A Beginner's Guide", Tata McGraw-Hill Edition [2001]

[5] Greg Buczck , "ASP.Net Developer's Guide", Third Edition, Tata McGraw-Hill Edition [2008].

[6] MichealOtey, DenielleOtey, "The Complete Reference ADO.Net", Third Edition, Tata McGraw-Hill Edition [2003].

[7] Dr.ShahramKhosravi, "ASP.Net Ajax Programmer's Reference", Second Edition, Wrox Publications [2007].

[8] J.Ambrose Little, Jason Beres, Grand Hinkson, Devin Rader, Joe Croney, "Silverlight 3 Programmer's Reference", Orely Publications[2009].