

Monitoring File System for Windows

Information Security

Prof. Nirmala Shinde¹

Assistant Professor: Department of Computer Engineering
K. J. Somaiya College of Engineering
Mumbai, India
e-mail: nirmalashinde@somaiya.edu

Prof. Smita Sankhe²

Assistant Professor: Department of Computer Engineering
K. J. Somaiya College of Engineering
Mumbai, India
e-mail: smitasankhe@somaiya.edu

Abstract— Every organization's asset is its data and data are stored in files which are maintained by file systems. Therefore, it is an important role of an organization to keep its File System secure. There is huge amount of changes that are made on daily basis in these files by different users. Hidden among these changes can be the few that are illegitimate and can cause harm to organization. So, File System Monitoring becomes necessary.

While many such monitoring tools are available for UNIX/Linux systems^[1], very little is done for Windows system. We have developed a File System Monitoring application for Windows operating system which monitors auditing of file systems – specifically, you want to know who read, modified, deleted or created files in a shared area.

While there were many options available for implementing such an application, the most appropriate way of doing so is by exploiting native compatibility of C.

Keywords- Interop, File Monitoring, Logs.

I. INTRODUCTION

A file system or filesystem is used to control how data is stored and retrieved. Without a file system, information placed in storage medium would be one large body of data with no way to tell where one piece of information stops and the next begins. By separating the data into pieces and giving each piece a name, the information is easily isolated and identified. Taking its name from the way paper-based information systems are named, each group of data is called a "file". The structure and logic rules used to manage the groups of information and their names are called a "file system". Most modern windows architecture was shown in figure 1.

Windows Architecture

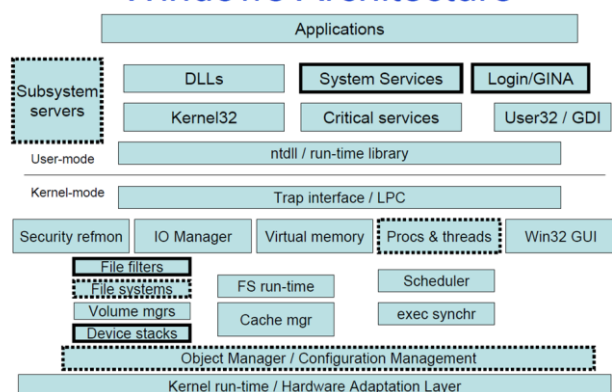


Figure 1 Windows Architecture

These files typically contain user data and application data, organization's important information such as trade secrets, system executables and databases. Changes to configurations, files and file attributes across the enterprise are

common, but hidden within a large volume of daily changes can be the few that impact file or configuration integrity. There are many motives for altering files. Intruders could modify system databases and programs to allow future access. Logs could be removed to cover their tracks. These modifications can heavily affect the integrity of the company and cause harmful damage to the company's data and assets. Therefore, the administrator needs to closely track activities of users who are using the system. File System monitoring is very critical for enterprises to monitor for attempted, unauthorized or unexpected modifications. It gives the system administrator the valuable data for processing and maintaining the system which are called Logs. The chosen Files and Folders are monitored by system and upon detecting changes; detailed event summary of those changes is logged. This will help administrator to Know Who Made What Changes when and where. The administrator can approve or disapprove changes as desired. Secure drive is a host-based intrusion detection system which is targeted for the use of organizations and/or institutions for performing File System monitoring on Windows System and reversing any changes that can cause damage. This software can be applied on local drive as well as shared drive like in the situations where certain sensitive files are being simultaneously accessed by many employees of the organization and the administrator wants to keep track of the changes made by all the employees and at the same time prevent any faulty/unethical manipulations to the files. Determining which files to monitor is a challenging question. Different types of files that can be monitored are configuration files, log files, digital keys and credentials and content driven files. Monitoring all files will create huge amount of data and

thus waste storage space. Hence, the administrator needs to carefully choose the directories and files to monitor. The software will detect the events of creating, renaming, deleting or changing file attributes such as last access, last write and log the detail summary of event. The logs of each file or directory will be saved as report which can be accessed by admin whenever required. The reports can also be printed and stored in pdf formats. Admin can approve or disapprove changes from logs based on whether the change was desired or not. This means that the changes of users will not be reflected permanently.

When the user tries to modify some file, it will get modified temporarily for the user but in actual, the file will not be affected unless the respective generated log gets approved by administrator. Content level changes, that is changes made within the file will also be detected and logged with detailed summary including who made what changes to which section or line number. The administrator can revoke or commit these changes. The trace of this software will be hidden from the Task Manager, that is it will work at kernel level, and hence the users will not know that they are being tracked. This will prevent the users from shutting down the software and covering their track. The software will be implemented using .Net Software Framework by Microsoft. It provides the best platform today for delivering Windows software. It has Native Interoperability with Windows Operating System. We have used C# for development which is general purpose; object-oriented programming language within .Net Framework as it provides native support with windows operating system.

II. ARCHITECTURE

Information Security of File System is functionally divided into two parts as shown in figure 2: Monitoring and Reporting. Monitoring deals with detecting the changes that are performed on the files whereas reporting deals with notifying the administrator about the changes that have taken place.

A. Monitoring

The system aims to monitor in real time access to sensitive files and folders stored on Windows System such as creating files in a directory, deleting, renaming and change in contents of the file. The change monitoring is divided into file level changes and content level changes.

• File Level Changes

All the basic file level operations such as creating, deleting and renaming a file are monitored. These changes can be accepted/rejected later by the administrator. In order to allow the functionality of bringing back the deleted file (in case of disapproval), all the files are backed up to a different directory. These basic file level changes are detected using the FileSystemWatcher class. The System.IO.FileSystemWatcher component class can be used in .NET applications to watch for changes in a specified directory. One can watch for changes in files and subdirectories of the specified directory. These

changes can be observed on a local computer, a network drive, or a remote computer.

We have used File System Watcher^[4] to track the changes in a specified directory. It works by creating component to watch files on local computer, network drive or remote computer. It has a "Filter" property to specify which files to monitor, it can be kept blank to watch all the files. We can select specific types of events to monitor on files and directories. It has an "Include Subdirectory" check option to enable monitoring events on sub directories. File System Watcher also monitors events on hidden files.

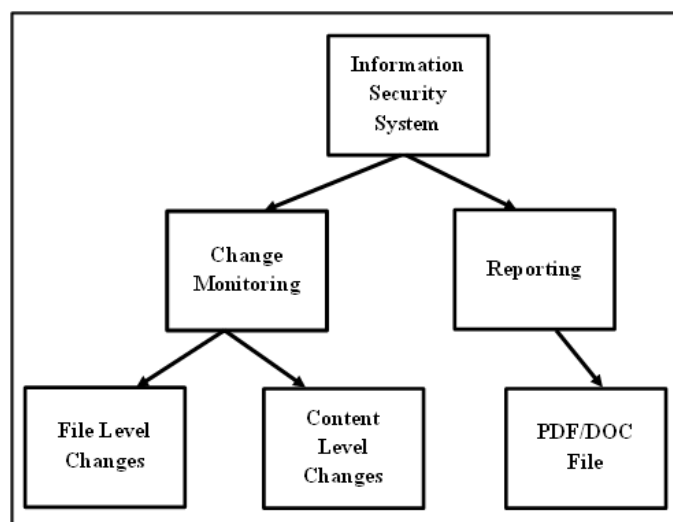


Figure 2 Information Security

• Content Level Changes

The main aim of this project lies in this part as there are a lot of changes made within the files and many users collaborate on the same file and make multiple changes, making it important to keep a track of the content level change activities done by the users. The project accomplishes this content level change monitoring with the help of Office Interop API. Microsoft Office Interop^[7] is an API compatible with MS Office products through programming. We can access various Office features like adding/ removing a word, formatting/ generating tables and reports, etc. Interop has a Revisions interface whose objects store the attributes of revisions made in the word file. We can accept or reject changes by accessing revisions through objects by calling the Revisions.Accept() or the Revisions.Reject() methods^[8].

For the content level changes in the excel files, there is no Revision Class available. So, the List Changes on New Sheet feature is used to keep a track of the changes made within the Work Book. All the modifications made in the cells across the sheets within that workbook are listed cell wise in the new sheet that has the following details: Old Value; New Value; Cell; Date-Time; User, Type of the change, etc. If a change made in a particular cell has to be rejected, it is done by selecting the cell-change record and programmatically set the value of that cell to the old value using the Interop.

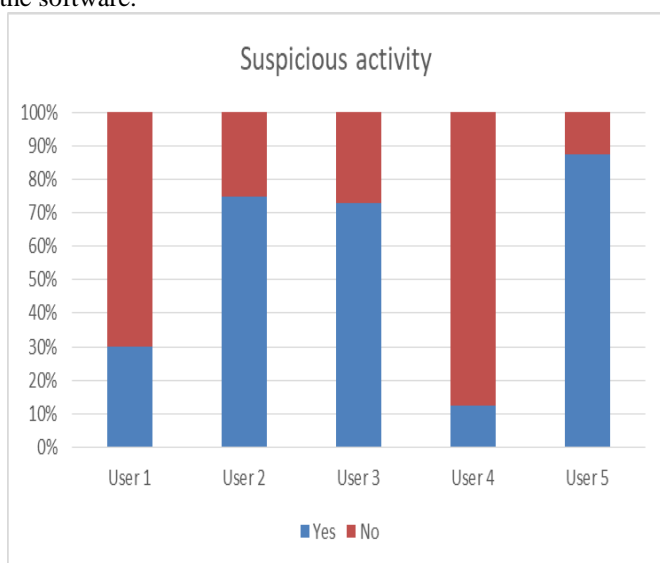
B. Reporting

The changes done by the users are detected and are notified to the administrator in the Reporting area of the system. File level changes and content level changes are displayed

separately. These changes can be accepted/rejected by the administrator. Once the changes are accepted/ rejected the records of those changes are removed from the reporting area. Changes done in multiple selected files are displayed at the same place which makes it convenient for tracking. All the reports can be saved in pdf or doc format.

III. RESULTS

Information Security successfully detects the content level changes on Word documents and Excel sheet using Interop's functionality and file level changes on all the files under selected directory using FileSystemWatcher. To overcome manual approval and disapproval of each user's modification always, user activities can be recorded and extracted from the application. This can be done by generating User specific reports in the same way the software is generating File specific reports. User specific reports records user actions like types of files accessed, types of changes made on file, and number of changes approved and disapproved by admin for that user. These reports can be analysed to infer suspicious users, for example, there could be a policy that states the user having his most modifications disapproved can be labelled suspicious. And such a user can be denied access to file structure to prevent further harm and admin overhead. Different other types of analysis can be exercised like which type of the file is most accessed, which kind of operation is frequently performed by user, number and type of changes that are mostly approved or disapproved, etc. Below are two examples of analysis carried out over static data extracted manually from the software.

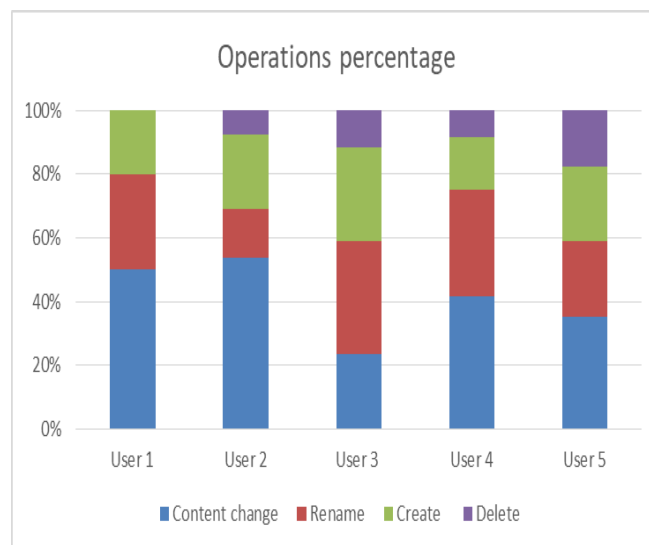


Graph. 1: Approval and Disapproval percentage of changes per user

Graph 1 shows the percentage of approval and disapproval of changes performed on the Y-axis and 5 users on the X-axis. Through this we can infer that the changes made by User 1 and User 4 have been disapproved more than they have been approved. Thus, these users are prone to be involved in suspicious activities. The managers can take actions such as restricting their rights to perform sensitive changes.

Graph 2 shows the percentage of each type of change done by the users on the Y-axis and 5 users on the X-axis. With the help of this we can infer that which user performs what kind of operation frequently. For example, the User 2 performs

highest percentage of content level changes. And User 1 has performed zero deletion operations.



Graph. 2: Types of changes done by user

The current data is static and manually recorded from software for Word and Excel file structures only. Further, the software can be expanded for other types of file systems whose data can also be recorded and analyzed in a similar manner.

IV. CONCLUSION

Future scope for the file system will be ownership rights can be monitored for file and directory, here the File Contents are tracked which allows the details of the actual changes made to a text-based configuration file. Rather than text based file image contented file can be monitored.

REFERENCES

- [1] Gene H. Kim, Eugene H. Spafford, "he design and implementation of tripwire: a file system integrity checker," ACM Conference, 1994.
- [2] Jean-Marc Boucqueau, "Digital Rights Management," IEEE . 2006=2012.
- [3] Tripwore Product.
<https://www.tripwire.com/products/tripwire-file-integrity-manager/>
- [4] FileSystemWathcer
[https://msdn.microsoft.com/en-us/library/system.io.filesystemwatcher\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.filesystemwatcher(v=vs.110).aspx)
- [5] File Integrity Monitoring and its Need
<https://www.alienvault.com/blogs/security-essentials/what-is-file-integrity-monitoring-and-why-you-need-it-part-2>
- [6] Existing File System Monitoring For Windows
<https://www.raymond.cc/blog/3-portable-tools-monitor-files-folders-changes/>
- [7] Interop Namespace
<https://msdn.microsoft.com/en-us/library/microsoft.office.interop.word.aspx?f=255&MSPPErrror=-2147217396>
- [8] Revision Interface in C#
- [9] <https://msdn.microsoft.com/en-us/library/microsoft.office.interop.word.revisions.aspx>
- [10] Mayur Mehta, "Design and Implementation of a File System Integrity Monitoring System", March 2016