



# **File Integrity Monitoring System**

*Version 4.0*

## **User Manual**



## The Development Team

This product was developed by a team of 3rd-year B.Tech Computer Science students at **VIT Chennai**, comprising **Aakash R**, **Subramanian Nachiappan**, and **Mizba J**, under the guidance and supervision of **Dr. Subbulakshmi T**.

## License and Usage

The File Integrity Monitor Project is an open source application. We welcome and encourage contributions from the community to help improve the File Integrity Monitor Project. If you wish to contribute, please contact the development team at [securazeta@gmail.com](mailto:securazeta@gmail.com). We strive to maintain a positive and inclusive collaborative environment in the File Integrity Monitor Project. The File Integrity Monitor Project and its contributors retain ownership of their respective contributions.

The File Integrity Monitor Project is provided "as is" without any warranties or guarantees of any kind. The project does not guarantee the accuracy, reliability, or security of the software, and users are solely responsible for their usage of the project. The project and its contributors shall not be liable for any damages or losses arising from the use of the software.

By using the File Integrity Monitor Project, you acknowledge and agree to these terms and conditions. Thank you for your contribution and support in making the project a success!

## Table of Content

<b>Chapter 1: Introduction</b>	<b>4</b>
1.1 Product Description	4
1.2 Product Users	4
1.3 Purpose Statement	5
1.4 Document Usage Requirements	6
1.5 Problem Reporting Instructions	7
<b>Chapter 2: Scope of Product</b>	<b>8</b>
<b>Chapter 3: Features</b>	<b>8</b>
<b>Chapter 4: User Guide</b>	<b>10</b>
4.1 Caution and Warning	10
4.2 Environment Compatibility	10
4.3 Installation and Setup	11
4.4 User Instructions	11
4.5 Related Information	15
<b>Chapter 5: Errors And Recovery</b>	<b>16</b>
<b>Chapter 6: Appendix</b>	<b>17</b>
3.1 Architecture	17
3.2 Version History	18

## **Chapter 1 - Introduction**

### **1.1 Product Description**

Introducing our advanced File Integrity Monitoring system that ensures the integrity of your directories and files. With our product, you can easily monitor changes to your directory and perform log analysis on the changes logged. Our product starts by creating a log file of the directory of interest, which contains four comma separated values: file name, status, time, and suspicion check. The status is determined using a hashing mechanism that compares and concludes if there has been an addition, modification, or deletion of files. Suspicion is determined using two methods: VirusTotal to check for suspicious hashes and PE file module to analyze executable files.

In addition, our product allows for manual log analysis using a regex-based filter, so you can acquire relevant log data that meets your needs. You can also request a copy of selected records to be emailed to you. Overall, our File Integrity Monitoring system provides a comprehensive solution for monitoring and ensuring the integrity of your directories and files. With advanced features and easy-to-use tools, our product is designed to meet your specific needs and protect your data from any unwanted changes or modifications.

### **1.2 Product Users**

The File Integrity Monitoring system is designed for users who need to monitor and ensure the integrity of their directories and files. This includes a wide range of users, such as IT administrators, security professionals, compliance officers, and system analysts.

- IT administrators can use this product to monitor changes made to the system and ensure that only authorized changes are made. They can

also use it to track system performance and troubleshoot any issues that may arise

- Security professionals can use this product to detect any unauthorized modifications or tampering with sensitive data. They can also use it to identify any suspicious activity or potential threats to the system.
- Compliance officers can use this product to ensure that their organization is compliant with relevant regulations and standards. They can use the monitoring system to track any changes made to sensitive data and ensure that data integrity is maintained.
- System analysts can use this product to analyze log data and identify trends and patterns in system performance. This can help them identify areas where improvements can be made and optimize system performance.

Overall, the File Integrity Monitoring system is a valuable tool for any user who needs to monitor and ensure the integrity of their directories and files, regardless of their specific role or industry.

### **1.3 Purpose Statement**

This manual is required for the using this product as it helps the users understand how to use the features of the File Integrity Monitoring system effectively. The manual provides detailed instructions on how to install and set up the product, how to monitor changes in your directories and files, how to analyze logs, environment compatibility, usage instructions and all other related information . It also includes guidelines on how to configure the product to meet specific needs and how to use advanced features. The user can also have insight about the architecture and working of the product.

The users can also know about the typically intended applications of the product include monitoring changes in critical directories and files of servers, ensuring compliance with regulatory requirements, detecting unauthorized access and modification of files in enterprise environments, and protecting sensitive data from unwanted changes or unauthorized access. Many other intended applications such as forensic investigations to track changes made to files and directories over time.

## 1.4 Document Usage Description

The document usage description for the File Integrity Monitoring system provides an overview of the document's purpose and structure, as well as guidance on how to use it effectively. The document is divided into several chapters, each with a specific focus.

- Chapter 1 introduces the product and provides essential information such as the intended audience, applicability, and problem reporting instructions. This chapter provides users with a high-level understanding of the product and what to do in case of any issues.
- Chapter 2 discusses the scope of the document, outlining the areas that the document covers and what it does not. This chapter ensures that users understand the limitations of the document and the product.
- Chapter 3 describes the features of the product, providing a detailed overview of what the product can do. It serves as a reference for users to understand the capabilities of the product and how to utilize them.
- Chapter 4 is the user guide section, which contains instructions for using the product. This chapter includes critical information such as caution and warning instructions, environment compatibility, installation and setup instructions, usage instructions, and related information. The user guide section provides step-by-step guidance on how to use the product effectively and optimize its features.
- Chapter 5 is focused on errors and recovery. It outlines how to identify issues that may arise when using the product and provides guidance on how to address those issues.
- Chapter 6 is the appendix section, which includes additional information on the product's architecture and version history. This

section is beneficial for users who want a deeper understanding of how the product works.

Overall, the document usage description emphasizes that each section of the document is interconnected, and users should refer to multiple sections to ensure they have a comprehensive understanding of the product. It also highlights the importance of following the problem reporting instructions provided in Chapter 1 to report any issues or problems encountered while using the product.

## 1.5 Problem Reporting Instructions

If you encounter any problems or issues with our File Integrity Monitoring system, please report them to us immediately. This will allow us to investigate and address the issue promptly, ensuring that you have the best possible experience with our product.

To report a problem, please email us at [securazeta@gmail.com](mailto:securazeta@gmail.com). In your email, please provide a detailed description of the issue you are facing, including any error messages or other relevant information. If possible, please include screenshots or logs that can help us to diagnose the problem more quickly.

Once we receive your email, we will investigate the issue and provide you with a solution or workaround as soon as possible. We appreciate your feedback and will work diligently to ensure that any issues are resolved promptly and to your satisfaction.

## Chapter 2 - Scope of Product

The advanced File Integrity Monitoring system can be used to monitor changes made to a directory and perform log analysis to detect suspicious activities in namely three ways - check for suspicious hashes, use the PE file module to analyse executable files and allow user to perform a manual log analysis using a regex-based filter. Lastly, it allows users to request a copy of selected records to be sent via email for future reference. Its advanced features help protect user data by alerting them timely.

Its user base can be tech professionals like IT administrators, security professionals, compliance officers, and system analysts or general public with an experience with computers and understanding of logs. Its applications include monitoring changes made to the system and ensuring authorization, track system performance, troubleshooting, keep track of tampering with sensitive data and detect potential threats, ensure compliance with standards and ensure data integrity. The field of interest include but are not limited to forensic domain, employee performance tracking and improve business.

## Chapter 3- Features

The File Integrity Monitoring system includes several key features that allow you to monitor changes to your directories and files and detect any suspicious activity. Some of the features of the product include:

- 1. Directory monitoring:** The product allows you to monitor changes to specific directories of interest and track any changes made to files within those directories.
- 2. Hash-based integrity checking:** The system uses hashing mechanisms to compare files and determine whether any changes have been made to the contents of the file.



- 3. Suspicious activity detection:** The product includes two methods of detecting suspicious activity - using Virus Total to check for suspicious hashes and using a PE file module to analyze executable files.
- 4. Log analysis:** The system logs all changes made to files in the monitored directories, allowing you to perform log analysis on the changes logged for the directory.
- 5. Regex-based filtering:** The product allows for manual log analysis using a regex-based filter to acquire relevant log data.
- 6. Notification system:** The product sends copies of selected records to the user on request.
- 7. Customization:** The product can be customized to meet specific needs and configured to monitor only certain types of files or directories.
- 8. Reporting:** The system generates reports on file changes and suspicious activity, which can be used for compliance purposes.
- 9. Scalability:** The product is scalable and can be used to monitor changes to files in large enterprise environments.

These features make the File Integrity Monitoring system an effective tool for monitoring changes to your directories and files and protecting your data from unauthorized additions, modifications or deletions.

## Chapter 4: User Guide

### 4.1 Caution and Warning

Upon the first run of the exe file, three files – ‘app.log’, ‘alert.log’ and ‘hashes.pkl’ are created automatically. **Do not modify, delete, rename, or relocate** these files. Failing to do so would result in unexpected and erroneous execution of the product.

Deletion of the ‘alert.log’ file would result in loss of file integrity log data. Relocation would create a new ‘alert.log’ resulting in loss of log history. Renaming would result in loss of log results during manual log analysis using regular expressions.





 alert.log	07-04-2023 19:14	Text Document	1 KB
 app.log	07-04-2023 19:14	Text Document	1 KB
 driver.exe	07-04-2023 14:26	Application	20,883 KB
 hashes.pkl	07-04-2023 19:14	PKL File	1 KB

Fig.1.File required-File Integrity Monitor

### 4.2 Environment Compatibility

- **Hardware:** Processor: 1 gigahertz (GHz) or faster processor or SoC. RAM: 1 gigabyte (GB) for 32-bit or 2GB for 64-bit. Hard disk space: 16GB for 32-bit OS 20GB for 64-bit OS
- **Operating System:** Windows 10 and above
- **Software:** System admin access to view file log in the terminal.
- **Network:** After installation, the tool require access to the internet.
- **Devices:** The tool can be run on PCs (desktops and laptops).
- **Mobile:** The current version of the tool is not compatible with any mobile platforms.
- **Versions:** The tool’s final version is compatible with Windows 10 and above.

## 4.3 Installation and Setup

To use the file integrity monitor just download the executable 'driver.exe' from the File-Integrity-Monitor github repository. The latest release of the product executable is available in the executable folder of the master branch in the github repository. Use the following link to access the github repository - <https://github.com/aakashr02/File-Integrity-Monitor>

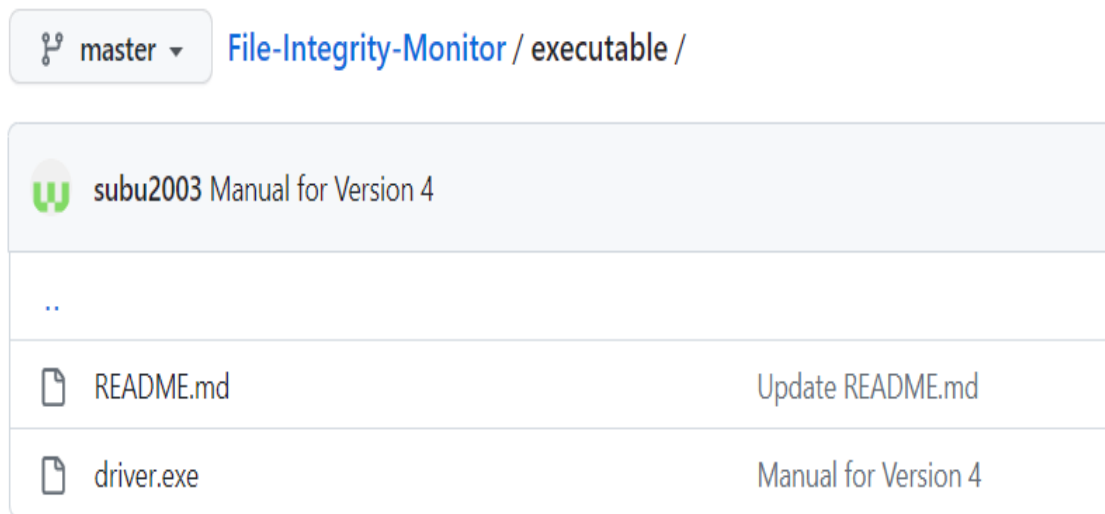
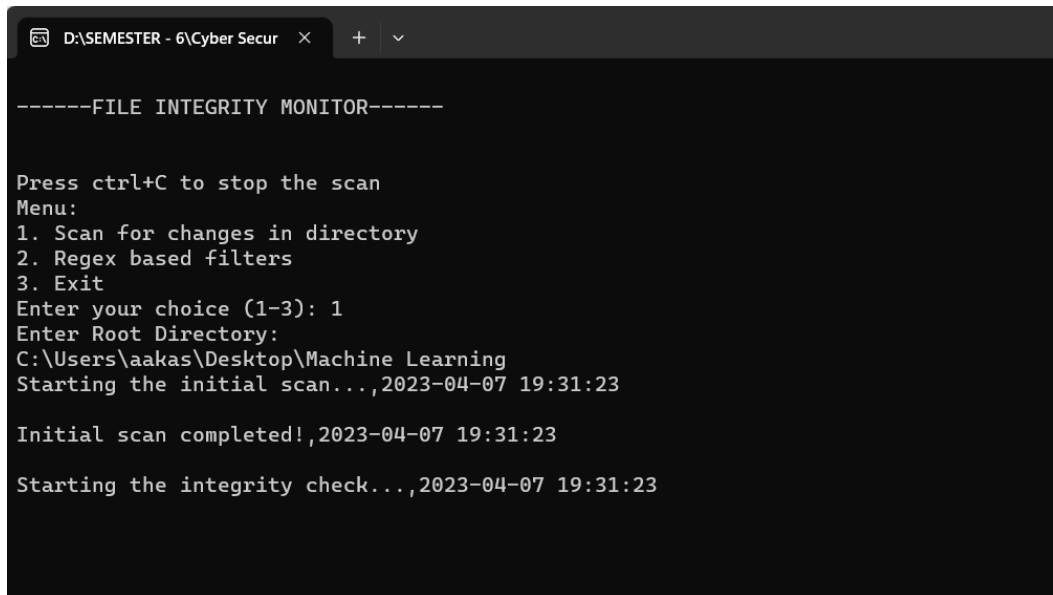


Fig.2. Executable File-File Integrity Monitor

## 4.4 User Instructions

### 4.4.1 Running a File Integrity Check

1. To run a scan session and perform a file integrity check, double click on the 'driver.exe' file to run it.
2. In the menu that appears enter 1 to scan a directory for changes.
3. Next specify the path of the directory that you want to scan and hit 'Enter'. Make sure the path is valid and exists.
4. Now the tool is ready to capture the changes in the directory tree specified



```
-----FILE INTEGRITY MONITOR-----

Press ctrl+C to stop the scan
Menu:
1. Scan for changes in directory
2. Regex based filters
3. Exit
Enter your choice (1-3): 1
Enter Root Directory:
C:\Users\alakas\Desktop\Machine Learning
Starting the initial scan...,2023-04-07 19:31:23

Initial scan completed!,2023-04-07 19:31:23

Starting the integrity check...,2023-04-07 19:31:23
```

Fig.3. Entering the path of Directory to be Monitored

5. To stop the scan, press 'ctrl+c' and terminate. The menu is displayed again.

#### 4.4.2 Interpreting Log Results

1. The changes in the directory including file addition, modification and deletions are stored in the 'alert.log' file.
2. The 'alert.log' has four columns. The first column specifies the type of change detected (add/change/remove).
3. The second column stores the filename including its complete path.
4. The third column stores the suspicion flag for the change captured in case of file addition and file content modification only. This is not applicable to file deletions.
5. The fourth column captures the timestamp of the change that happened in the directory and its sub directories.
6. The log results are displayed at the end of the scan session.

```
-----Summary of the Scan Session-----
```

Status	File Name	Flag	Time
change	/hiring.csv	Safe	2023-04-07 19:42:16
change	/requirements.txt	Safe	2023-04-07 19:42:33
add	/New Text Document.txt	Safe	2023-04-07 19:42:54
add	/hello.txt	Safe	2023-04-07 19:42:59
remove	/New Text Document.txt	-	2023-04-07 19:42:59
change	/hello.txt	Safe	2023-04-07 19:43:10
remove	/hello.txt	-	2023-04-07 19:43:20

Fig.4. Summary of Scan Session

#### 4.4.3 Detect Suspicious

1. Run a live scan on a directory to capture suspicious changes.
2. The flag attribute in the log result displayed at the end of the scan session depicts the suspicion status of a change occurred during the scan session.
3. A 'safe' flag indicates that the particular change captured is not suspicious while a 'suspicious' flag indicates that it is suspicious.
4. Now the tool is ready to capture the changes in the directory tree specified

```
-----Summary of the Scan Session-----
```

Status	File Name	Flag	Time
add	/malicious.php	Safe	2023-04-07 20:47:53
add	/New Text Document.txt	Safe	2023-04-07 20:48:04
change	/New Text Document.txt	Safe	2023-04-07 20:48:20
change	/malicious.php	Suspicious	2023-04-07 20:49:13

Fig.5. Detecting Suspicious Files

5. To stop the scan, press 'ctrl+ c' and terminate. The menu is displayed again.

#### 4.4.4 Perform Manual Log Analysis

1. The Regular Expression based filter provides a manual log analysis feature.
2. To perform log analysis enter 2 in the when the choice is asked.
3. Next specify your email address and hit enter.
4. Then enter the desired regular expression.
5. The filtered log records based on the regular expression supplied are displayed.
6. Here are a few examples –

##### a. Look for file additions

```
Enter your choice (1-3): 2
Enter Email Address: aakashrajan0202@gmail.com
Enter the Regular Expression: add

-----Filtered Records-----
Add : 2      Change : 0      Remove : 0
```

Status	File Name	Flag	Time
add	C:\Users\aaakas\Desktop\Machine Learning\malicious.php	Safe	2023-04-07 20:47:53
add	C:\Users\aaakas\Desktop\Machine Learning\New Text Document.txt	Safe	2023-04-07 20:48:04

##### b. Look for suspicious file changes

```
Enter your choice (1-3): 2
Enter Email Address: aakashrajan0202@gmail.com
Enter the Regular Expression: Suspicious

-----Filtered Records-----
Add : 0      Change : 1      Remove : 0
```

Status	File Name	Flag	Time
change	C:\Users\aaakas\Desktop\Machine Learning\malicious.php	Suspicious	2023-04-07 20:49:13

##### c. Look for changes on '.txt' files

```
Enter Email Address: aakashrajan0202@gmail.com
Enter the Regular Expression: \.txt

-----Filtered Records-----
Add : 1      Change : 1      Remove : 1
```

Status	File Name	Flag	Time
remove	C:\Users\aaakas\Desktop\Machine Learning/sample.txt	-	2023-04-07 20:46:43
add	C:\Users\aaakas\Desktop\Machine Learning\New Text Document.txt	Safe	2023-04-07 20:48:04
change	C:\Users\aaakas\Desktop\Machine Learning\New Text Document.txt	Safe	2023-04-07 20:48:20

### 4.4.5 Exporting Log Data

1. To export all log records or particular records using regular expression based filter, enter 2 in the choice menu.
2. Next enter your email address to which the log data is to be exported.
3. Then enter the desired regular expression or enter `^.*$` to export all log records.
4. A success message is displayed upon a successful export. If the export fails, check your internet connectivity.

```

Enter your choice (1-3): 2
Enter Email Address: aakashrajan0202@gmail.com
Enter the Regular Expression: ^.*$

-----Filtered Records-----
Add : 2      Change : 4      Remove : 1

+-----+-----+-----+-----+
| Status | File Name | Flag | Time |
+-----+-----+-----+-----+
| change | C:\Users\aa...Machine Learning\malicious.php | Safe | 2023-04-07 20:46:22 |
| change | C:\Users\aa...Machine Learning\malicious.php | Safe | 2023-04-07 20:46:28 |
| remove | C:\Users\aa...Machine Learning/sample.txt | - | 2023-04-07 20:46:43 |
| add | C:\Users\aa...Machine Learning\malicious.php | Safe | 2023-04-07 20:47:53 |
| add | C:\Users\aa...Machine Learning/New Text Document.txt | Safe | 2023-04-07 20:48:04 |
| change | C:\Users\aa...Machine Learning/New Text Document.txt | Safe | 2023-04-07 20:48:20 |
| change | C:\Users\aa...Machine Learning\malicious.php | Suspicious | 2023-04-07 20:49:13 |
+-----+-----+-----+-----+
Email has been sent successfully!!!!!!!!!!!!

```

Fig.6. Regex Based Filter Records

## 4.5 Related Information

### 1. Regular Expression -

Regular Expressions are often used alongside manual log file analysis. They provide an easy and convenient way to filter records from very large log data. File Integrity Monitor provides a regular expression based filtering option to allow users to perform manual log analysis to identify suspicious changes and irregular trends on a directory. The section 6.3 of the Appendix consists of characters and symbols used to specify regular expressions.

## 2. Attack and Intrusion Detection -

Attack and Intrusion Detection is another activity performed alongside file integrity monitoring. A File Integrity Monitor provides information on changes captured on a particular directory tree in the form of a log. This log data can be used as a basis to identify and detect attacks and intrusions into the file system using more sophisticated tools and approaches.

## Chapter 5: Errors And Recovery

### 1. Could not send mail

**Error** - This error occurs when the File Integrity Monitor is unable to export the log data to your mail. It occurs mostly because the application is not connected to the internet.

**Recovery** - Check if your device is connected to the internet. Check the strength of the internet connection.

### 2. Undetermined Suspicion Flag -

**Error** - The Suspicion flag for a file addition or modification is marked 'Undetermined' when the application is unable to determine the suspicion status for the particular file change. This is because the application is unable to connect to the VirusTotal Database to fetch the results .

**Recovery** - Check if your device is connected to the internet. Check the strength of the internet connection.

### 3. Abrupt Termination of application -

**Error** - The application may terminate and close abruptly when the regular expression entered for filtering out the log data is invalid. The log data is not lost at any cost.



**Recovery** - Open the application again and enter 2 in the choice option to perform regular expression based filter. This time enter a valid regular expression. Refer the section 6.3 of Appendix on how to use regular expressions.

#### 4. No changes detected -

**Error** - There may be a delay to capture the changes in the directory but would not go undetected, provided the path to the directory is valid and exists. This delay may be a result of computing and checking the file hashes especially when the directory tree is large and has a large amount of files. However, if the directory path specified does not exist or is invalid, then the changes are not captured.

**Recovery** - If the changes in the directory are not captured during the scan session, check if the directory path specified is correct. If not, terminate the scan session by pressing 'ctrl+c' and re-run the scan again by specifying the correct directory name and path.

## Chapter 6 - Appendix

### 6.1. Architecture

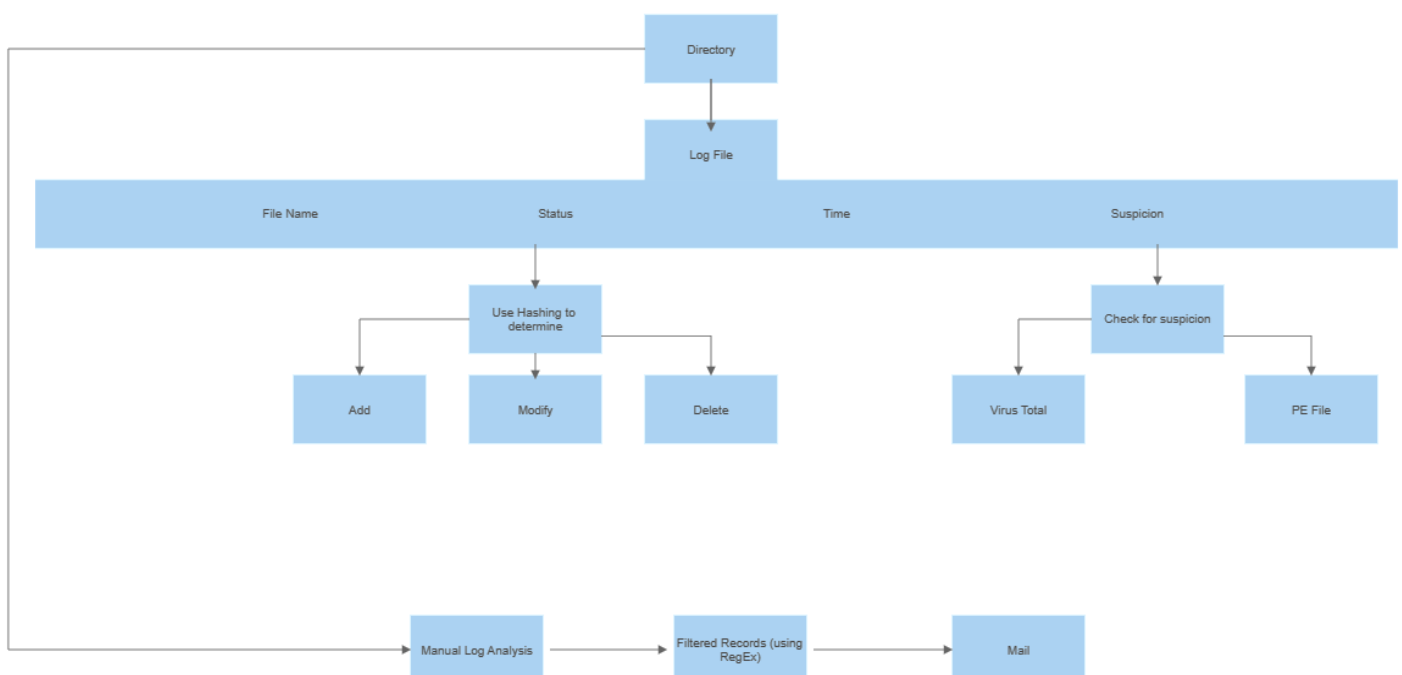


Fig.8. Architecture Diagram

- **Directory:** It is used to store, organise and separate files in a given subspace. The path of the directory that needs to be scanned is entered in the tool.
- **Log File:** The changes in the directory are stored in 'alert.log' file which has 4 columns namely filename which specifies the file of interest, status states whether the scan has been completed or still in process, timestamp determines the time at which the change was detected and suspicion detects suspicious activities in case of additions and modifications to files alone.
- **Virus Total:** Suspicious hashes can be determined using VirusTotal.
- **PE File Module:** It is used to check for suspicion in executable files (.exe).
- **Manual Log Analysis using RegEx based filter:** A RegEx expression can be specified in the terminal to filter files that satisfy certain parameters for further analysis.
- **Mail:** The user can acquire a copy of filtered records from the log on request.

## 6.2 Version History

The product has been developed as four versions. Each version's key features and bug fixing as been discussed below:-

### 6.2.1 Version 1

This version is the basic version of the application. Which computes the hash of all the files in the given directory and its sub directories and stores the hashes for continuously monitoring any changes in the file hashes. If any changes is detected in the file hashes, changes are recorded to 'alert.log'. It also has additional feature to detect what type of changes which is made to file (Add/Modify/Delete). But this version had a bug as it only detected for modification in file and it was not able to detect the addition and deletion of files.

#### Key Features-

- Computing the hashes of all files in given directory and its sub directory
- Detect for changes in file content in directory
- Recording the changes in 'alert.log' along with the time of detection
- Recording the start and end time of the scan session in 'app.log'

### 6.2.2 Version 1.1

#### Bug Fixing-

Bug in detecting the type of change(Add/Remove) was fixed

#### Key Features-

- Classifying the type of change(Add/Modify/Delete)
- Accept the user choice of directory
- Terminate the scan using Keyboard Interrupt

### 6.2.3 Version 2

#### Key Features-

- Providing a regex based filter for user to filter the log changes according to their needs
- Providing a menu for user for navigating between the File Monitoring and regex

### 6.2.3 Version 3

#### Key Features-

- Flagging whether the file is suspicious using two methods-
  - PE File Module
  - Virus Total API

### 6.2.4 Version 3.1

#### Bug Fixing-

- Bug in the regex based output fixed
- Error in formatting the results fixed

#### Key Features-

- Accepting the regex expression from the user for filtering the records

- Tabulating the results of the Scanned session and regex using pretty table module
- Formatting the output further for better user convenience

## 6.2.5 Version 4

### Key Features-

- Creating a separate log file for storing the user filtered records
- Writing the filtered records into 'result.log' for communicating the results
- Mailing the report to the user for their further references with a appropriate styling and formatting

### Bug Fixing-

- Addition of Try and Catch Block to handle the errors while sending the file hash to the Virus Total API system due to disconnectivity from the Network
- Addition of Try and Catch Block to handle the errors due to network disconnectivity while mailing the log file details to the user

## 6.3 Regular Expression

Anchors	
<code>^</code>	Start of line +
<code>\A</code>	Start of string +
<code>\$</code>	End of line +
<code>\Z</code>	End of string +
<code>\b</code>	Word boundary +
<code>\B</code>	Not word boundary +
<code>&lt;</code>	Start of word
<code>&gt;</code>	End of word

Sample Patterns	
<code>([A-Za-z0-9-]+)</code>	Letters, numbers and hyphens
<code>(\d{1,2}\V\d{1,2}\V\d{4})</code>	Date (e.g. 21/3/2006)
<code>([^\s]+(?:=\.(jpg gif png))\.\s{2})</code>	jpg, gif or png image
<code>(^[1-9]{1}\$ ^[1-4]{1}[0-9]{1}\$ ^50\$)</code>	Any number from 1 to 50 inclusive
<code>(#(?:[A-Fa-f0-9]{3}([A-Fa-f0-9]{3})?)</code>	Valid hexadecimal colour code
<code>((?=[*\d])(?=[*a-z])(?=[*A-Z]).{8,15})</code>	8 to 15 character string with at least one upper case letter, one lower case letter, and one digit (useful for passwords).
<code>(\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6})</code>	Email addresses
<code>(\&lt;\/?[^\&gt;+])\&gt;)</code>	HTML Tags

Character Classes	
<code>\c</code>	Control character
<code>\s</code>	White space
<code>\S</code>	Not white space
<code>\d</code>	Digit
<code>\D</code>	Not digit
<code>\w</code>	Word
<code>\W</code>	Not word
<code>\xhh</code>	Hexadecimal character hh
<code>\Oxxx</code>	Octal character xxx

POSIX Character Classes	
<code>[[:upper:]]</code>	Upper case letters
<code>[[:lower:]]</code>	Lower case letters
<code>[[:alpha:]]</code>	All letters
<code>[[:alnum:]]</code>	Digits and letters
<code>[[:digit:]]</code>	Digits
<code>[[:xdigit:]]</code>	Hexadecimal digits
<code>[[:punct:]]</code>	Punctuation
<code>[[:blank:]]</code>	Space and tab
<code>[[:space:]]</code>	Blank characters
<code>[[:cntrl:]]</code>	Control characters
<code>[[:graph:]]</code>	Printed characters
<code>[[:print:]]</code>	Printed characters and spaces
<code>[[:word:]]</code>	Digits, letters and underscore

Assertions	
<code>?=</code>	Lookahead assertion +
<code>?!</code>	Negative lookahead +
<code>?&lt;=</code>	Lookbehind assertion +
<code>?!= or ?&lt;!</code>	Negative lookbehind +
<code>?&gt;</code>	Once-only Subexpression
<code>?()</code>	Condition [if then]
<code>?() </code>	Condition [if then else]
<code>?#</code>	Comment

Note	
Note	These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.

Quantifiers	
<code>*</code>	0 or more +
<code>*?</code>	0 or more, ungreedy +
<code>+</code>	1 or more +
<code>+?</code>	1 or more, ungreedy +
<code>?</code>	0 or 1 +
<code>??</code>	0 or 1, ungreedy +
<code>{3}</code>	Exactly 3 +
<code>{3,}</code>	3 or more +
<code>{3,5}</code>	3, 4 or 5 +
<code>{3,5}?</code>	3, 4 or 5, ungreedy +

Special Characters	
<code>\</code>	Escape Character +
<code>\n</code>	New line +
<code>\r</code>	Carriage return +
<code>\t</code>	Tab +
<code>\v</code>	Vertical tab +
<code>\f</code>	Form feed +
<code>\a</code>	Alarm
<code>[\b]</code>	Backspace
<code>\e</code>	Escape
<code>\N{name}</code>	Named Character

String Replacement (Backreferences)	
<code>\$n</code>	nth non-passive group
<code>\$2</code>	"xyz" in <code>/^(abc(xyz))\$/</code>
<code>\$1</code>	"xyz" in <code>/^(?:abc)(xyz)\$/</code>
<code>\$`</code>	Before matched string
<code>\$'</code>	After matched string
<code>\$+</code>	Last matched string
<code>\$&amp;</code>	Entire matched string
<code>\$_</code>	Entire input string
<code>\$</code>	Literal "\$"

Ranges	
<code>.</code>	Any character except new line ( <code>\n</code> ) +
<code>(a b)</code>	a or b +
<code>(...)</code>	Group +
<code>(?:...)</code>	Passive Group +
<code>[abc]</code>	Range (a or b or c) +
<code>[^abc]</code>	Not a or b or c +
<code>[a-q]</code>	Letter between a and q +
<code>[A-Q]</code>	Upper case letter + between A and Q +
<code>[0-7]</code>	Digit between 0 and 7 +
<code>\n</code>	nth group/subpattern +

Note	
Note	Ranges are inclusive.

Pattern Modifiers	
<code>g</code>	Global match
<code>i</code>	Case-insensitive
<code>m</code>	Multiple lines
<code>s</code>	Treat string as single line
<code>x</code>	Allow comments and white space in pattern
<code>e</code>	Evaluate replacement
<code>U</code>	Ungreedy pattern

Metacharacters (must be escaped)		
<code>^</code>	<code>[</code>	<code>.</code>
<code>\$</code>	<code>{</code>	<code>*</code>
<code>(</code>	<code>\</code>	<code>+</code>
<code>)</code>	<code> </code>	<code>?</code>
<code>&lt;</code>	<code>&gt;</code>	

Note	
Note	Items marked + should work in most regular expression implementations.

Fig.8. Regex Syntax