A Mini Project Report On

File Integrity Monitoring System

A Dissertation submitted in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science &Engineering (CYBER SECURITY)

Submitted by

(Student Name)	**(Roll No)**
P.Sai Charan	**(21H51A6269)**
A.Manaswini	**(21H51A6294)**
Md Sameer Ali	**(21H51A62C5)**

Under the esteemed guidance of
T.Krishnaveni
Assistant professor, CSE (Cyber Security)

Department of Computer Science & Engineering (CYBER SECURITY)

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous)
(NAAC Accredited with ‘A+’ Grade & NBA Accredited) (Approved by AICTE, Permanently Affiliated to JNTU Hyderabad)
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD-501401
2021-25

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(CYBER SECURITY)

CERTIFICATE

This is to certify that the Mini Project – 1 report entitled **“File Integrity Monitoring System”** being submitted by **P.Sai Charan (21H51A6269), A.Manaswini (21H51A6294), Md Sameer Ali (21H51A62C5)** in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering (Cyber Security)** is a record of bonafide work carried out his/her under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

T.krishnaveni	**Dr. R.Venkateswara Reddy**
(Assistant Professor)	**(Assoc. Professor & HOD)**
Dept. of CSE-Cyber Security	**Dept. of CSE-Cyber Security**

ACKNOWLEDGMENT

With great pleasure I want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

I am grateful to **T.Krishnaveni ,** Assistant Professor, Dept of Computer Science and Engineering-Cyber Security for his valuable suggestions and guidance during the execution of this project work.

I would like to thank **Dr. R.Venkateswara Reddy**, Head of the Department of Computer Science and Engineering-Cyber Security, for his moral support throughout the period of my study in CMRCET.

I am highly indebted to **Major Dr. V.A. Narayana**, Principal CMRCET for giving permission to carry out this project in a successful and fruitful way.

I would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

Finally, I express my sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

P.Sai Charan	**(21H51A6269)**
A.Manaswini	**(21H51A6294)**
Md. Sameer Ali	**(21H51A62C5)**

DECLARATION

We hereby declare that results embodied in this Report of Project on **“File Integrity Monitoring System”** are from work carried out by using partial fulfillment of the requirements for the award of B. Tech degree. We have not submitted this report to any other university/institute for the award of any other degree.

NAME	ROLL NO	SIGNATURE
P.Sai Charan	(21H51A6269)	
A.Manaswini	(21H51A6294)	
Md. Sameer Ali	(21H51A62C5)	

ABSTRACT

In the contemporary landscape of cybersecurity, the File Integrity Monitoring (FIM) system stands as a stalwart guardian, addressing the escalating challenges posed by unauthorized access and tampering of critical files within computer systems and networks. This abstract delves into the essence of FIM, elucidating its significance, functionalities, and overarching contributions to the security posture of organizations. File Integrity Monitoring operates on the principle of upholding the integrity and security of digital assets. It entails continuous surveillance of files, directories, and system configurations, with the primary goal of swiftly detecting and responding to any deviations from the established baseline. FIM systems serve as vigilant sentinels, providing real-time alerts and responses to a spectrum of security incidents, from insidious malware infiltrations to potential insider threats. The multifaceted role of FIM extends beyond the immediate realm of cybersecurity. By ensuring the sanctity of data and configurations, FIM systems contribute to regulatory compliance, aligning organizations with industry standards and frameworks. This compliance is essential for instilling confidence among stakeholders and demonstrating a commitment to the safeguarding of sensitive information.

Table Of Content

CHAPTER 1

1.INTRODUCTION

In the digital age, where the importance of data integrity and cybersecurity is more critical than ever, organizations face a constant challenge in protecting their sensitive information from unauthorized access and tamper . In the ever-evolving landscape of cybersecurity, safeguarding sensitive data and critical system files is paramount. File Integrity Monitoring (FIM) systems have emerged as indispensable tools in the arsenal of cybersecurity measures. These systems play a crucial role in detecting and responding to unauthorized changes within a computer system or network, helping organizations fortify their defenses against cyber threats.

File integrity monitoring is predicated on the fundamental principle of ensuring the integrity and security of files and configurations. As organizations increasingly rely on digital data and information systems, the potential for unauthorized access, data breaches, and tampering grows. FIM systems act as vigilant guardians, continuously monitoring files, directories, and system configurations to identify any deviations from the established baseline.

The primary objective of a File Integrity Monitoring system is to detect, alert, and respond to unauthorized alterations in real-time. By maintaining a watchful eye on critical files, FIM systems empower organizations to proactively address security incidents, ranging from malware infections to insider threats. This proactive stance is vital in a landscape where cyber threats are becoming more sophisticated and dynamic.

Beyond the realm of cybersecurity, FIM systems contribute to regulatory compliance by fulfilling requirements set forth by industry standards and frameworks. Many regulatory bodies mandate the implementation of file integrity monitoring to ensure the protection of sensitive information and to demonstrate a commitment to data integrity and security.

This introduction sets the stage for exploring the multifaceted role of File Integrity Monitoring systems in the realm of cybersecurity. From preventing unauthorized changes and data breaches to aiding in forensic analysis and ensuring compliance, FIM systems are integral components in the broader strategy of safeguarding digital assets and maintaining the trust of stake holders in an interconnected and digital world.

1.1 AIM

The aim of our project is to enhance the security posture of a system or network by detecting and responding to unauthorized changes, thereby protecting sensitive data, ensuring compliance, and maintaining system integrity.

1.2 SCOPE AND OBJECTIVES

The scope of File Integrity Monitoring is dynamic, adapting to the evolving landscape of cybersecurity. As threats become more sophisticated, FIM remains a critical component in the proactive defense against unauthorized changes and the preservation of data integrity.

The goal of this project is to protect sensitive files from unauthorized changes by keeping a log and sending real-time alerts to the administrator.

1.Advanced Detection Techniques:

Explore and implement advanced detection techniques, such as anomaly detection, behavior analysis, and machine learning, to enhance the FIM system's ability to identify both known and emerging threats.

2.Evasion Resistance:

Investigate methods to make FIM systems more resistant to evasion techniques employed by cyber adversaries. This includes studying polymorphic malware, encryption-based attacks, and other strategies used to bypass traditional FIM defenses.

3.Insider Threat Mitigation:

Develop mechanisms within the FIM system to effectively distinguish between legitimate changes made by authorized users and changes resulting from insider threats, whether intentional or unintentional.

CHAPTER 2

2.LITERATURE REVIEW

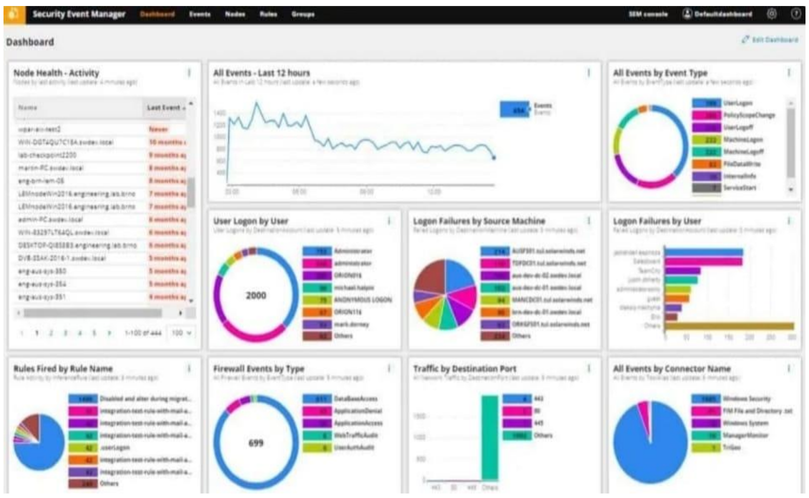
"Tripwire: A File System Integrity Checker" by Eugene H. Spafford (1992)
This seminal paper introduces Tripwire, a file integrity checking tool that detects unauthorized changes to files on a system, providing a foundation for modern file integrity monitoring systems.
"Integrity Measurement Architecture" by James Newsome, Dawn Song (2004)
The paper introduces the Integrity Measurement Architecture (IMA), a framework for ensuring the integrity of a system's files. It presents a design that allows for the measurement of various system components, providing a basis for creating a secure and trustworthy system through file integrity monitoring.
"AIDE - Advanced Intrusion Detection Environment" by Rami Lehti, et al,(2010)
The paper introduces AIDE, a robust and flexible file integrity checker. It discusses the design and features of AIDE , emphasizing its ability to detect and report unauthorized changes to files, making it a valuable tool for system administrators and security professionals.
"Osiris: An Open Source Host Integrity Monitoring System" by Martin Roesch (1999)
Summary: The paper introduces Osiris, an open-source host integrity monitoring system. Osiris is designed to monitor multiple hosts and detect changes to files by comparing cryptographic hashes. The paper discusses the system's architecture, capabilities, and its role in maintaining the integrity of files in a distributed environment.

CHAPTER 3

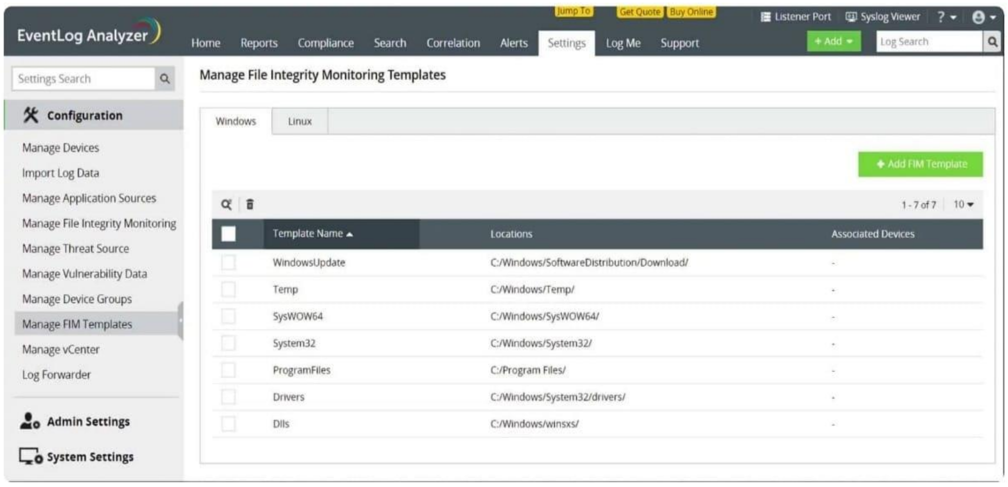
3.EXISTING SOLUTIONS

1.SolarWinds Security Event Manager :

Manages logs as well as providing security scans for threats. Log messages are drawn from all around the system so they can be pooled for searching and then stored in files. These files are important for re-searching and for compliance reporting and so they are protected against tampering.



2.ManageEngine EventLog Analyzer is a SIEM system that is very similar to the Solar Winds Security Event Manager . This package deploys extensivefile integrity monitoring and those files that have been marked as containing sensitive data get any changes made to them logged, which means that those changes can be reversed.



Manage File Integrity Monitoring Templates

Template Name	Locations	Associated Devices
WindowsUpdate	C:/Windows/SoftwareDistribution/Download/	-
Temp	C:/Windows/Temp/	-
SysWOW64	C:/Windows/SysWOW64/	-
System32	C:/Windows/System32/	-
ProgramFiles	C:/Program Files/	-
Drivers	C:/Windows/System32/drivers/	-
Dlls	C:/Windows/winsxs/	-

3. Miscellaneous websites and softwares:

Similar websites are widely available, but their trials are restricted. The service is paid for.

CHAPTER 4

4.PROPOSED SYSTEM

The proposed system will ensure the integrity and security of system files by verifying their integrity against known hashes or checksums. It will regularly scan the operating system files and compare them with a trusted database to detect any unauthorized modifications or tampering. The integrity checker will provide real-time alerts and notifications for any discrepancies found, enabling administrators or users to take immediate action.

1.Adding file path:

Add the path of file to a loaded base line to activate the file integrity checker

2.Checksum generation:

The file integrity monitoring system generates a checksum for the file and stores it in the local database server.

3.Regular scans:

It regularly scans the files with the previously stored checksums and compare them with a trusted database to detect any unauthorized modifications or tampering.

4.Real time alerts and notifications for any discrepancies found, enabling administrators or users to take immediate action.

4.1REQUIREMENT ANALYSIS

4.1.1Software Requirements

- Powerhell-GUI

4.1.2Hardware Requirements

- System 32 bit with 4 GB RAM

4.2 MERITS AND DEMERITS

MERITS:

1.Detecting a cyberattack-In the early stages of a complex cyberattack, cybercriminals may need to modify critical files related to the operating system or applications. Very sophisticated attackers may even alter the log files to conceal their activity. FIM tools, on the other hand, can detect such changes because they compare the current file to a baseline rather than simply reviewing the file logs.

2. Expediting threat detection, response and remediation-By detecting a threat early in the cyber kill chain, an organisation has a better chance of preventing a breach before it causes significant or costly damage. FIM significantly improves the organization's ability to detect attacks that other security measures may miss.

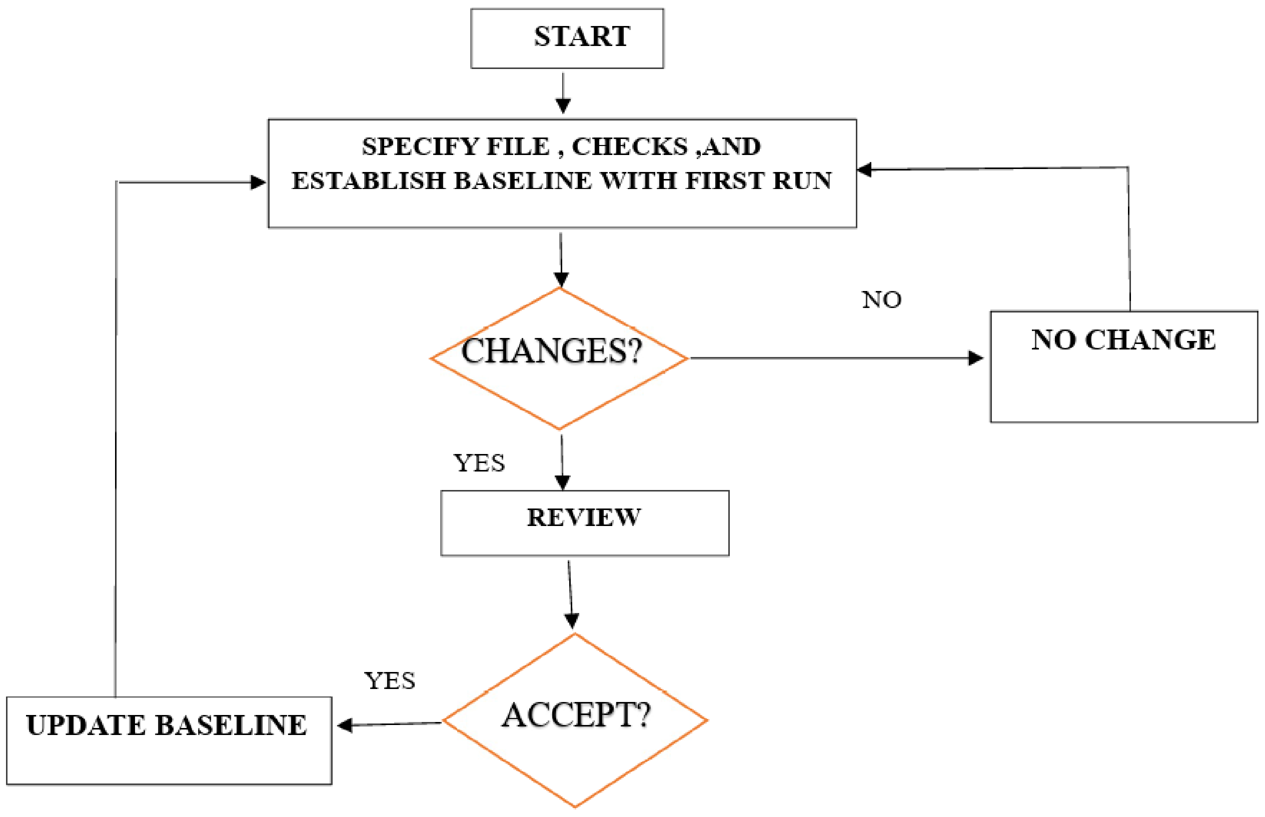
3. Identifying weaknesses within the IT infrastructure-While not malicious in nature, some changes made by administrators or employees may inadvertently expose the organisation to risk. FIM assists in identifying and correcting vulnerabilities before they are exploited by adversaries.

DEMERITS:

FIM software generates false positives. If the baseline is not properly configured, the FIM toolinterprets a legitimate change as an unauthorized one, generates a warning message, and its alerts become false positive.

CHAPTER 5

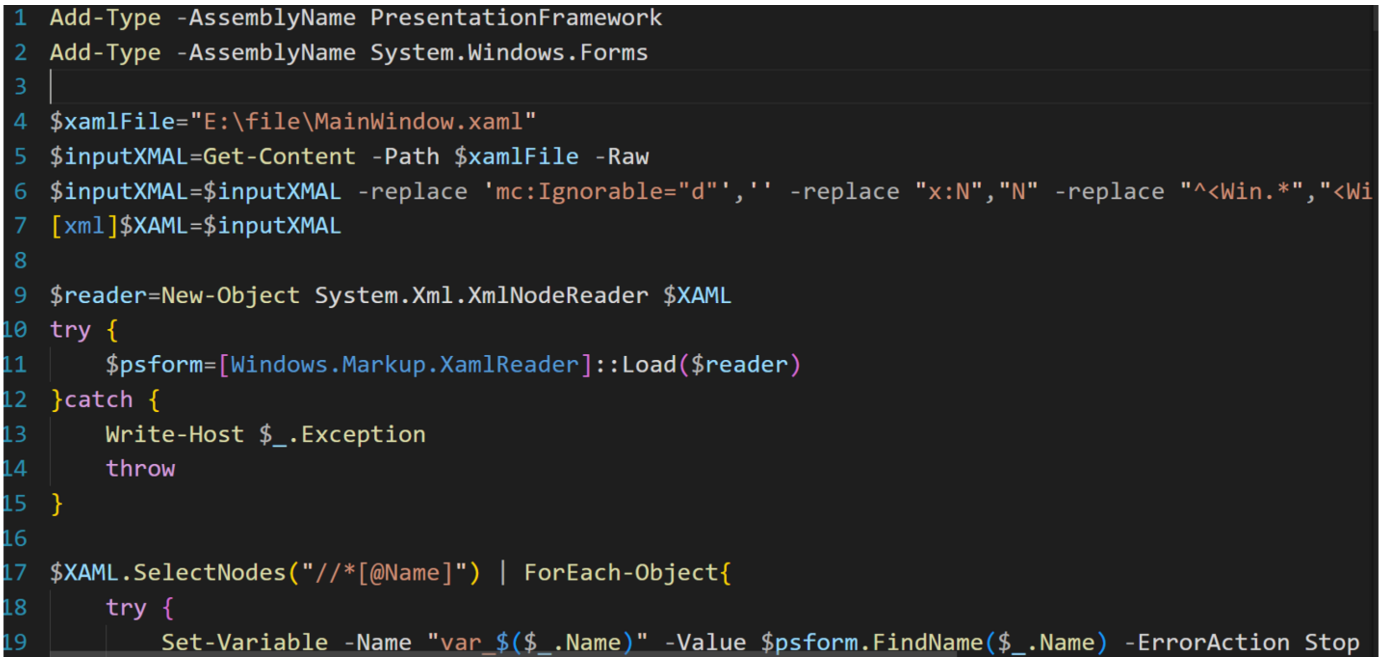
5.1. CONCEPTUAL DESIGN


START
SPECIFY FILE , CHECKS ,AND ESTABLISH BASELINE WITH FIRST RUN
CHANGES?
NO
NO CHANGE
YES
REVIEW
YES
ACCEPT?
UPDATE BASELINE

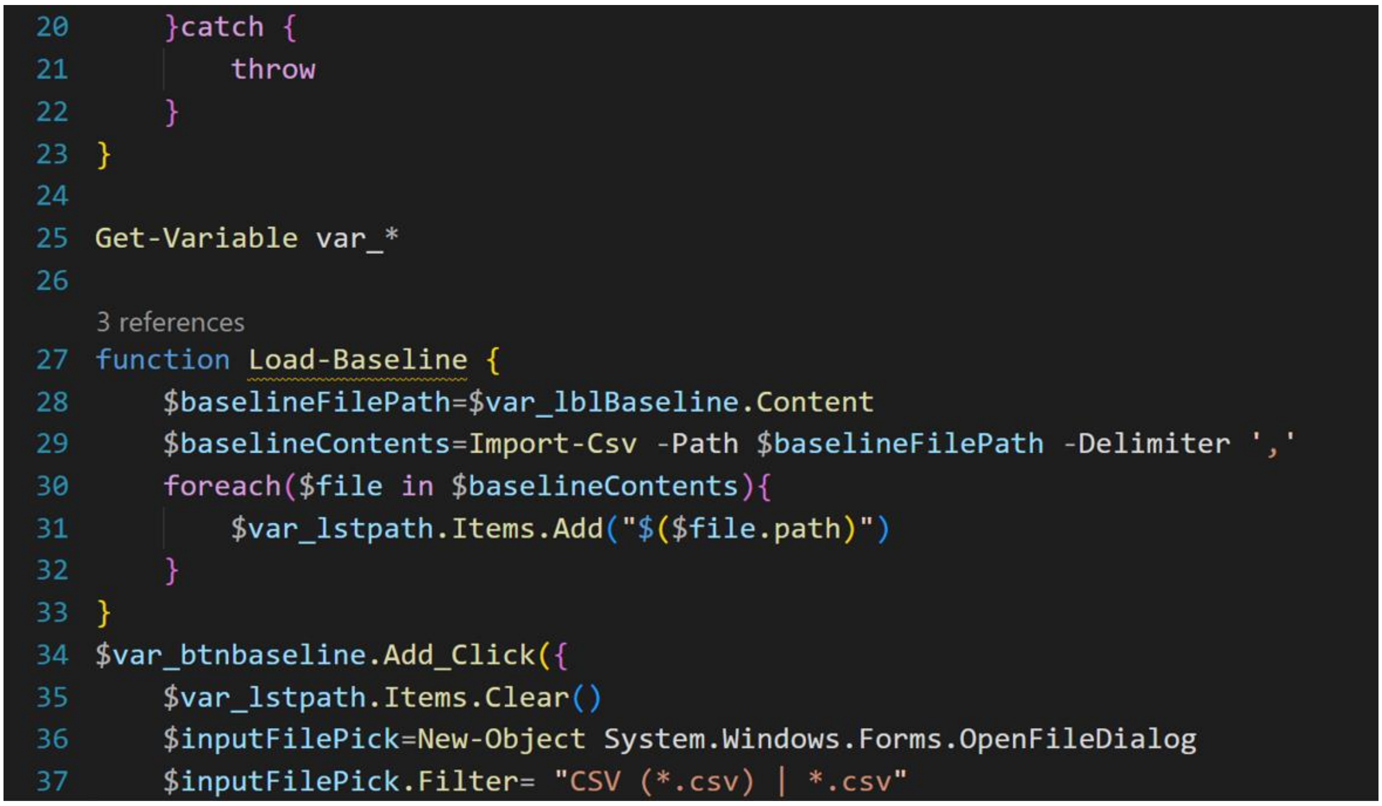
CHAPTER 6

6.IMPLEMENTATION

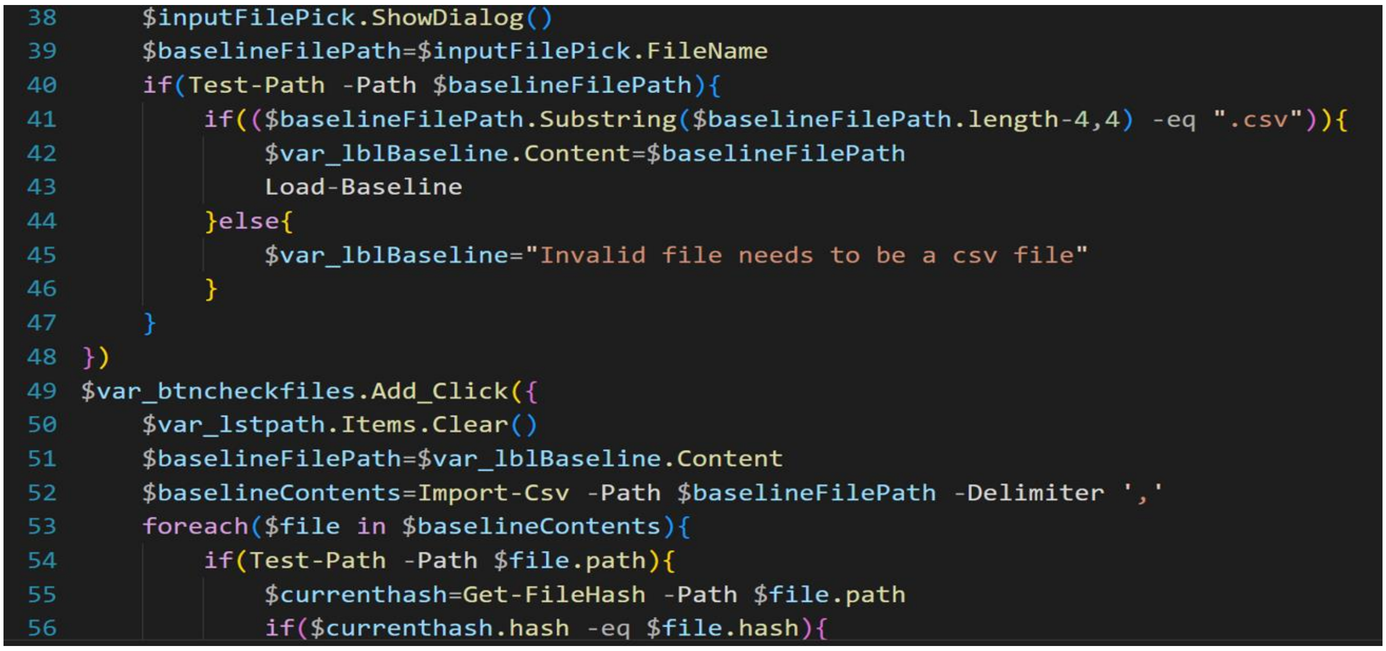
Power shell program



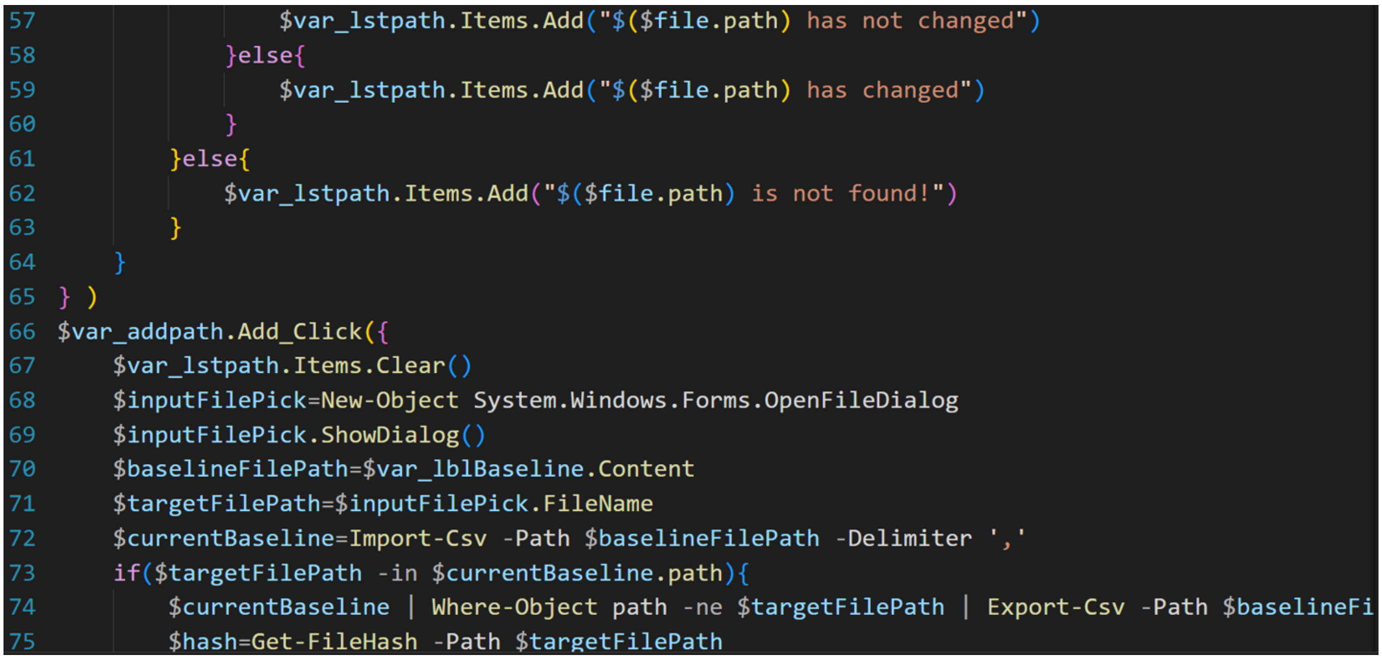
```
Add-Type -AssemblyName PresentationFramework
Add-Type -AssemblyName System.Windows.Forms

$xamlFile="E:\file\MainWindow.xaml"
$inputXMAL=Get-Content -Path $xamlFile -Raw
$inputXMAL=$inputXMAL -replace 'mc:Ignorable="d"','' -replace "x:N","N" -replace "^<Win.*","<Wi
[xml]$XAML=$inputXMAL

$reader=New-Object System.Xml.XmlNodeReader $XAML
try {
    $psform=[Windows.Markup.XamlReader]::Load($reader)
}catch {
    Write-Host $_.Exception
    throw
}

$XAML.SelectNodes("//*[@Name]") | ForEach-Object{
    try {
        Set-Variable -Name "var_$($_.Name)" -Value $psform.FindName($_.Name) -ErrorAction Stop
```



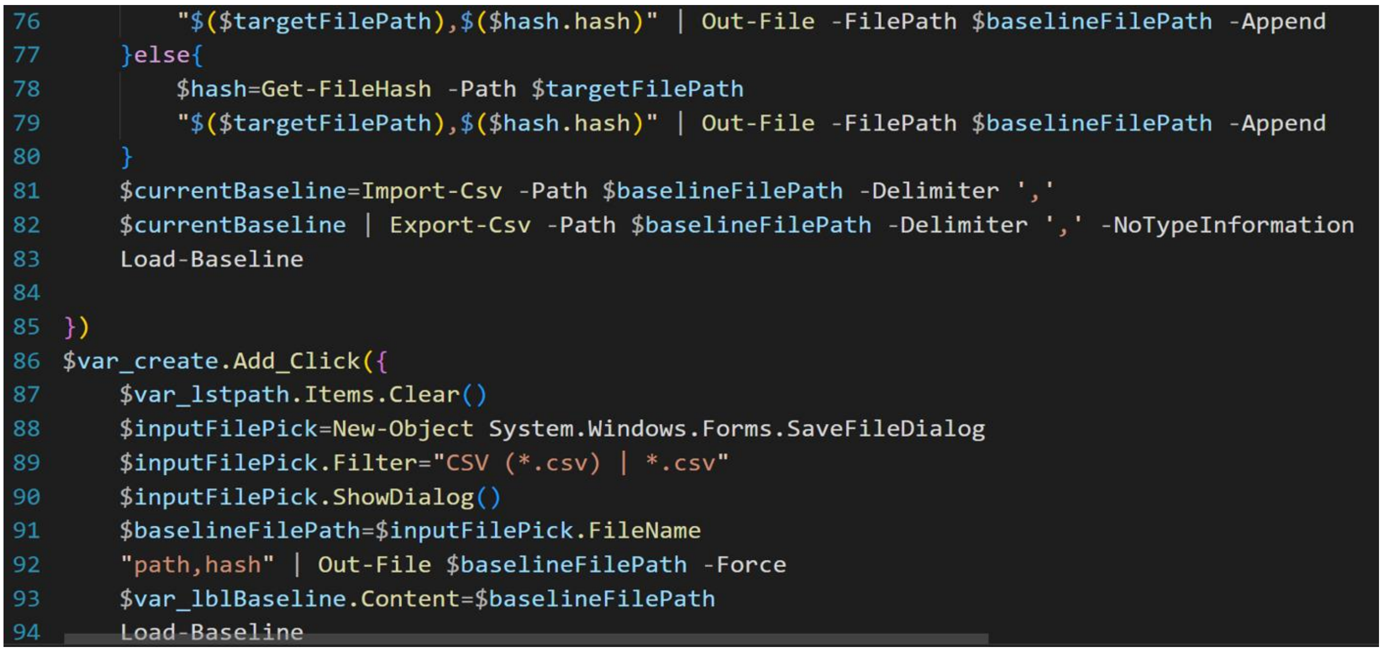
```
    }catch {
        throw
    }
}

Get-Variable var_*

3 references
function Load-Baseline {
    $baselineFilePath=$var_lblBaseline.Content
    $baselineContents=Import-Csv -Path $baselineFilePath -Delimiter ','
    foreach($file in $baselineContents){
        $var_lstpath.Items.Add("$($file.path)")
    }
}
$var_btnbaseline.Add_Click({
    $var_lstpath.Items.Clear()
    $inputFilePick=New-Object System.Windows.Forms.OpenFileDialog
    $inputFilePick.Filter= "CSV (*.csv) | *.csv"
```



```
    $inputFilePick.ShowDialog()
    $baselineFilePath=$inputFilePick.FileName
    if(Test-Path -Path $baselineFilePath){
        if(($baselineFilePath.Substring($baselineFilePath.length-4,4) -eq ".csv")){
            $var_lblBaseline.Content=$baselineFilePath
            Load-Baseline
        }else{
            $var_lblBaseline="Invalid file needs to be a csv file"
        }
    }
})
$var_btncheckfiles.Add_Click({
    $var_lstpath.Items.Clear()
    $baselineFilePath=$var_lblBaseline.Content
    $baselineContents=Import-Csv -Path $baselineFilePath -Delimiter ','
    foreach($file in $baselineContents){
        if(Test-Path -Path $file.path){
            $currenthash=Get-FileHash -Path $file.path
            if($currenthash.hash -eq $file.hash){
```

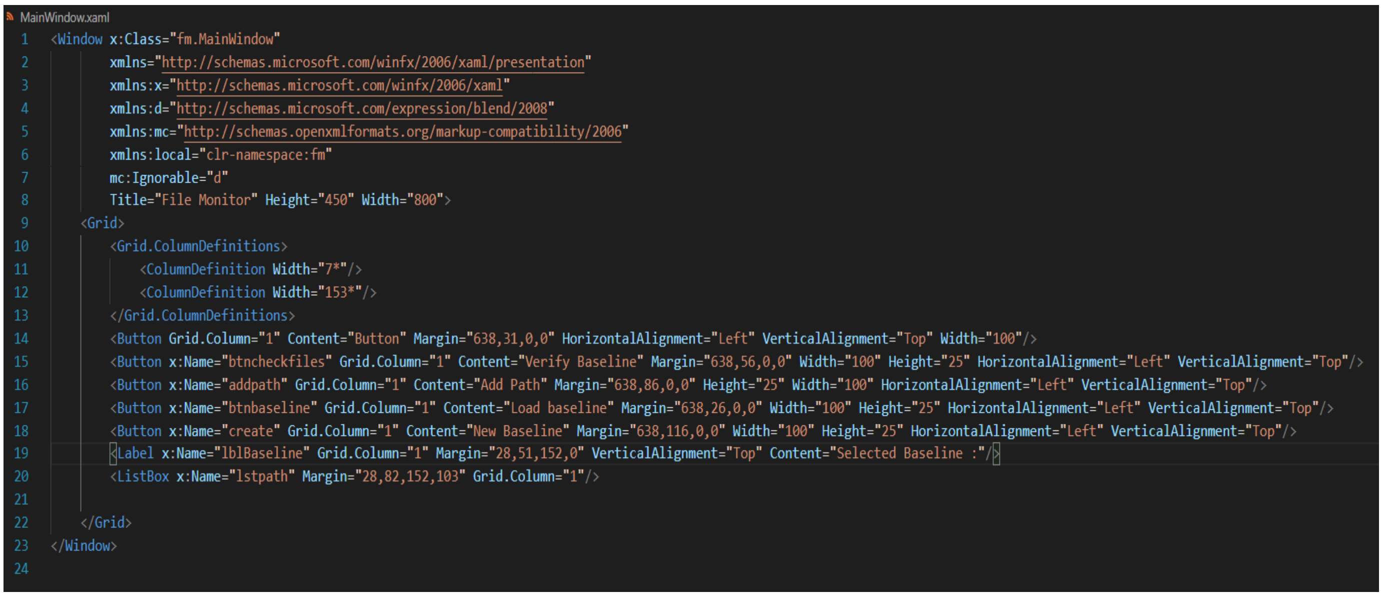


```
                $var_lstpath.Items.Add("$($file.path) has not changed")
            }else{
                $var_lstpath.Items.Add("$($file.path) has changed")
            }
        }else{
            $var_lstpath.Items.Add("$($file.path) is not found!")
        }
    }
} )
$var_addpath.Add_Click({
    $var_lstpath.Items.Clear()
    $inputFilePick=New-Object System.Windows.Forms.OpenFileDialog
    $inputFilePick.ShowDialog()
    $baselineFilePath=$var_lblBaseline.Content
    $targetFilePath=$inputFilePick.FileName
    $currentBaseline=Import-Csv -Path $baselineFilePath -Delimiter ','
    if($targetFilePath -in $currentBaseline.path){
        $currentBaseline | Where-Object path -ne $targetFilePath | Export-Csv -Path $baselineFi
        $hash=Get-FileHash -Path $targetFilePath
```



```
            "$($targetFilePath),$($hash.hash)" | Out-File -FilePath $baselineFilePath -Append
        }else{
            $hash=Get-FileHash -Path $targetFilePath
            "$($targetFilePath),$($hash.hash)" | Out-File -FilePath $baselineFilePath -Append
        }
        $currentBaseline=Import-Csv -Path $baselineFilePath -Delimiter ','
        $currentBaseline | Export-Csv -Path $baselineFilePath -Delimiter ',' -NoTypeInformation
        Load-Baseline

})
$var_create.Add_Click({
    $var_lstpath.Items.Clear()
    $inputFilePick=New-Object System.Windows.Forms.SaveFileDialog
    $inputFilePick.Filter="CSV (*.csv) | *.csv"
    $inputFilePick.ShowDialog()
    $baselineFilePath=$inputFilePick.FileName
    "path,hash" | Out-File $baselineFilePath -Force
    $var_lblBaseline.Content=$baselineFilePath
    Load-Baseline
```

XML FILE



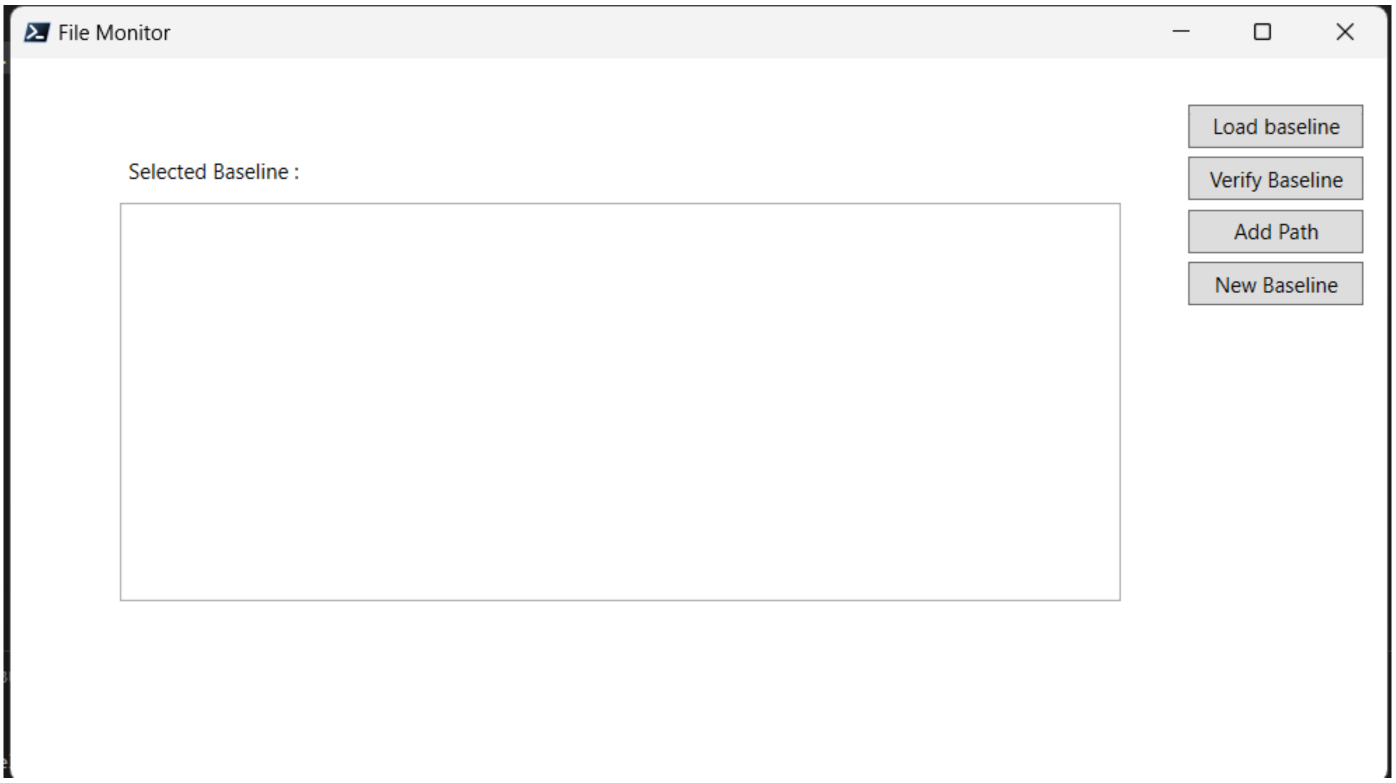
MainWindow.xaml

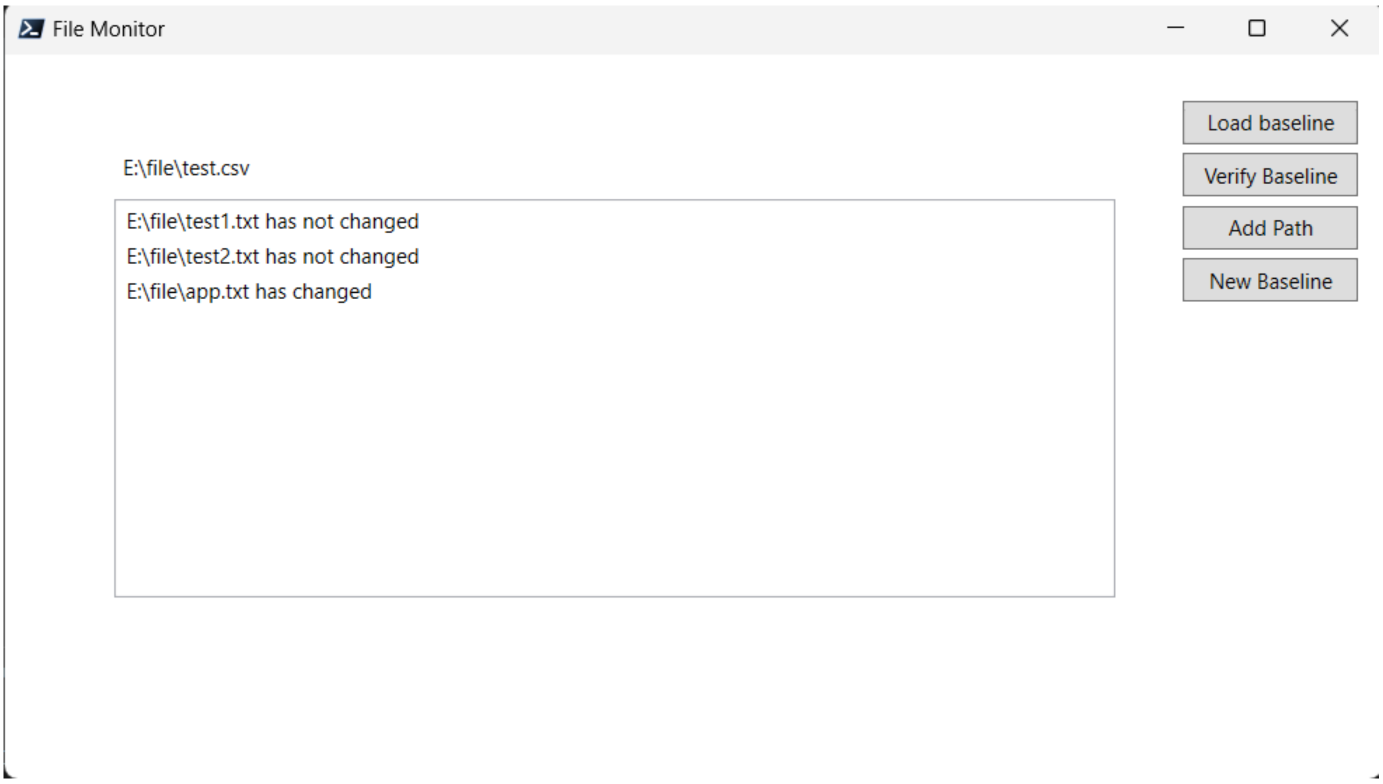
```
<Window x:Class="fm.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:fm"
        mc:Ignorable="d"
        Title="File Monitor" Height="450" Width="800">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="7*"/>
            <ColumnDefinition Width="153*"/>
        </Grid.ColumnDefinitions>
        <Button Grid.Column="1" Content="Button" Margin="638,31,0,0" HorizontalAlignment="Left" VerticalAlignment="Top" Width="100"/>
        <Button x:Name="btncheckfiles" Grid.Column="1" Content="Verify Baseline" Margin="638,56,0,0" Width="100" Height="25" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="addpath" Grid.Column="1" Content="Add Path" Margin="638,86,0,0" Height="25" Width="100" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="btnbaseline" Grid.Column="1" Content="Load baseline" Margin="638,26,0,0" Width="100" Height="25" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="create" Grid.Column="1" Content="New Baseline" Margin="638,116,0,0" Width="100" Height="25" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Label x:Name="lblBaseline" Grid.Column="1" Margin="28,51,152,0" VerticalAlignment="Top" Content="Selected Baseline :"/>
        <ListBox x:Name="lstpath" Margin="28,82,152,103" Grid.Column="1"/>

    </Grid>
</Window>

```

CHAPTER 7

7.RESULT

First we create a baseline to store the checksums of the selected files ,we can also add new file paths to the baseline and verify the integrity of the files.If the files remain same it displays unchanged ,if the data in a selected file is selected it displays file changed.





E:\file\test.csv

```
E:\file\test1.txt has not changed
E:\file\test2.txt has not changed
E:\file\app.txt has changed
```

CHAPTER 8

8.1.CONCLUSION:

In conclusion, File Integrity Monitoring (FIM) systems play a crucial role in fortifying the security posture of organizations by continuously monitoring and validating the integrity of critical files and system configurations. However, as with any security technology, FIM systems come with their set of strengths and limitations.

On the positive side, FIM systems offer real-time monitoring capabilities, enabling organizations to swiftly detect and respond to unauthorized changes. The establishment of a baseline through cryptographic hashing provides a reference point for identifying alterations, and the integration of threat intelligence feeds enhances the system's ability to identify emerging threats. The forensic capabilities of FIM systems also contribute valuable insights for post-incident analysis and investigations.

FIM Is an essential Security requirement. However, detecting any specific change is only the beginning of the process. To be effective, FIM solutions must distinguish between low-risk and high-risk change, integrate with other security solutions for log and security event management (including real-time alerts), and support a fully managed change history database

8.2.FUTURE ENHANCEMENTS

1. Add a sign up and login to account to save encrypted and decrypted data.

CHAPTER 9

9.REFERENCES

1. https://ieeexplore.ieee.org/document/7501350
2. https://ieeexplore.ieee.org/document/9736317
3. https://www.researchgate.net/publication/331258764_A_New_RealTime_File_Integrity_Monitoring_System_for_Windows-based_Environments
4. https://www.sciencedirect.com/science/article/pii/S0898122110000180