AUTOMATED EVENT LOG ANALYSIS AND ALERT SYSTEM VIA EMAIL

BY

MD. TAREK KHAN
ID: 191-15-2489

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering [Font-14]

Supervised By

Professor Dr. Touhid Bhuiyan
Professor
Department of CSE
Daffodil International University

Co-Supervised By

Mr. Afjal Hossan Sarower
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

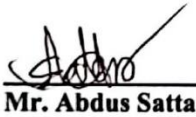
JULY 2024

APPROVAL

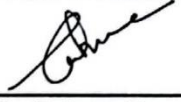
This Project titled "**AUTOMATED EVENT LOG ANALYSIS AND ALERT SYSTEM VIA EMAIL**", submitted by **Md. Tarek Khan** to the department of Computer Science And Engineering Daffodil International University, Has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and Approved as to its style and contents. The Presentation has been held on 13th July 2024.

BOARD OF EXAMINERS

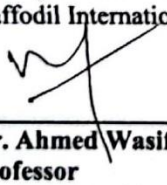
Dr. Md. Ismail Jabiullah(MIJ) — **Board Chairman**
Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University



Mr. Abdus Sattar (AS) — **Internal Examiner 1**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University



Ms. Zahura Zaman (ZZ) — **Internal Examiner 2**
Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University



Dr. Ahmed Wasif Reza (DWR) — **External Examiner**
Professor
Department of Computer Science and Engineering
East West University

DECLARATION

I hereby declare that, this project has been done by us under the supervision of **Dr. Touhid Bhuiyan, Professor.** Due to the leave of **Dr. Touhid Bhuiyan** sir, all aspects of my project implementation were guided by and the project report was checked by **Mr. Afjal Hossan Sarower** sir, Department of CSE Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Dr. Touhid Bhuiyan
Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Mr. Afjal Hossan Sarower
Lecturer (Senior Scale)
Department of CSE
Daffodil International University

Submitted by:

Md. Tarek Khan
ID: 191-15-2489
Department of CSE
Daffodil International University

iii

©Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Dr. Touhid Bhuiyan**, **Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Information Security*" to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Dr. Sheak Rashed Haider Noori Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

The analysis of Windows event logs forms a vital basis today in this fast-changing cyber landscape for obtaining control accompanied by integrity and security of IT ecosystems. Traditional manual log analysis is, however, time-demanding, prone to errors, and very often financially inaccessible to purchase expensive SIEM for small and medium-sized enterprises (SMEs). To address these challenges, the automated "Event Log Analysis and Alert System via Email" project is introduced as an innovative solution that might easy access to automatic log analysis. It uses Python scripting and Windows PowerShell commands to automate data collection, analysis, and distribution from event logs to facilitate efficient and timely monitoring and response to security events. Using open-source tools and technologies within the project offers a cost-effective solution comparing with expensive Security Information and Event Management (SIEM) systems. Upon the first execution, clients provide inputs like email, hours, and select storage options (email or Google Drive). The system stores these inputs for future use and executes the exe file automatically every 24 hours using Task Scheduler, Thus continuously collecting, storing, and sending log data without further user intervention. The system detect Critical, Warning, Error and AuditFailure Events and then create these log data as a CSV file for send email and stores in Google Drive. It can also store this log data in a central MySQL database on the phpmyadmin.co site to send to the dashboard using PHP. This project has a dashboard controlled by the SOC, or Security Administrator, which has a drop-down button where they can select EventIDs and get detailed information of how many hosts are affected by each EventID. This would significantly improve the speed and accuracy of identifying threats, troubleshooting, and incident response. It tremendously enhances the overall cybersecurity and system activity monitoring of an IT environment.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Introduction

IT environments are high-stake zones for today's digital ecosystems, and the security and integrity of these environments are the most important factors, anywhere. Windows event logs are an important part of cybersecurity, as these logs contain useful information that can help identify malicious activity on the system or may indicate a possible security breach and troubleshoot any issues. But this human processing of records is typically manual, time-consuming, and error-prone. In addition, installing and keeping standard Security Information and Event Management (SIEM) solutions running can be very costly, according to some companies and even small and medium-sized enterprises (SMEs). The project “Automated Event Log Analysis and Alert System via Email” seeks to address these issues through an innovative approach. This project captures the Windows event log data, analyzes the data, and spreads it out using automated Python scripting combined with Windows PowerShell commands. It also stored this log data into Google Drive and a centralized MySQL database hosted on the phpmyadmin.co site. Which is viewable through the web-based dashboard, which is maintained by the SOC or Security Administrator. The aim of this project is to make log analysis automation readily accessible to everyone by providing an accessible and affordable alternative to the expensive SIEM solutions equipped with a user-friendly dashboard to help all kinds of enterprises and IT environments.

1.2 Motivation

The primary motivation behind the project “Automated Event Log Analysis and Alert System via Email” is to adjustment the landscape of traditional cybersecurity log analysis. Given the nature of emerging cybersecurity threats against the backdrop of modern digital environments, maintaining the security and integrity of IT environments continues to be a challenge. Windows event logs are a critical component in the fight against such threats, providing invaluable data that can help in detecting and responding to potential security incidents and troubleshoot any issues. These logs represent a complete history of system

activities, errors, and warnings, serving as an incredibly useful resource for troubleshooting any problems that occur in IT environments.

However, the conventional manual methods of log analysis are riddled with inefficiencies, including time constraints and susceptibility to human error. Financial barriers render traditional Security Information and Event Management (SIEM) solutions inaccessible to small and medium-sized enterprises (SMEs).

In the end, what motivates this project is the belief that organizations of all kinds should be able to protect themselves against increasingly diverse and agile cyber threats by monitoring their windows event logs. This project takes a step towards leveling the playing field by providing an innovative avenue that makes it easier for pretty much any organization to be proactive in maintaining the security and integrity of their IT environment, free from the constraints of financial burden or technological limitations.

1.3 Rational of the Study

The study's rationale is based on the critical need to implement efficient and cost-effective solutions for analysis in modern IT event log environments. Modern IT environments pose complex and voluminous cybersecurity challenges; hence, organizations must be constantly under the threat of ever-changing cyber threats in terms of complexity and volume while monitoring and analyzing organizational security responses. Traditional manual methods of log analysis are time-consuming, error-prone, and they consume significant financial and human resources. This is particularly a burden for small to medium-sized enterprises.

To address the problems stated above, this study designs an automated system using Python scripting to run Windows PowerShell commands. The output is expected to automate the data collection, analysis, and dissemination of the Windows event log data. Automation brings a way to speed these processes up by lessening manual effort used in the analysis,

minimizing human error, and ensuring it gives a scalable and cost-effective alternative to expensive SIEM solutions.

Secondly, the tool is designed to bridge the present gap in the availability and use of advanced log analysis tools by SMEs to strengthen their cybersecurity without being laden with high financial investment burdens. It is systematized and sends alerts via email. It has centralized MySQL database storage available on phpmyadmin.co site, with additional data storage and backup in Google Drive. This enables a Security Operations Center (SOC) or security administrator to monitor and respond effectively to such critical, warning, and error events.

1.4 Related Questions

1. How do Windows event logs contribute to the identification and mitigation of security threats within IT environments?
2. What are the key limitations of traditional Security Information and Event Management (SIEM) solutions in addressing the needs of organizations, particularly small to medium-sized enterprises (SMEs)?
3. What potential benefits can be derived from automating the collection, analysis, and dissemination of event log data?
4. How can Python scripting and Windows PowerShell commands be effectively utilized to streamline event log analysis processes?
5. How can the accessibility and affordability of automated log analysis capabilities be improved for organizations with varying resources and capabilities?

1.5 Expected Output

The expected output of the "Automated Event Log Analysis and Alert System via Email" project will substantially enhance the log analysis and security monitoring capabilities of any organization, especially SMEs, which generally do not have the resources to purchase some expensive SIEM systems. It will automate the collection of event log data from Windows systems based on user defined parameters, such as specific time frames in hours and e-mail addresses of recipients. This eliminates manual log retrieval, hence minimizing human error. It generates a CSV file consisting of log data on the occurrence of critical, warning, error, and AuditFailure events. It sends them through email to SOC or Security Administrator, thus ensuring effective communication about critical security information in a timely way. It also stores the log data in a remote centralized MySQL database hosted on the phpmyadmin.co site, and on Google Drive. These are safe storage solutions that provide easy management and appropriate access. When the clients run the EXE for the first time, there will be a user-friendly interface where they can input their email address and the duration of log collection they prefer, either through email or Google Drive storage. These inputs will be stored for future use and run automatically without user interaction. The Task Scheduler shall be used to collect logs automatically and generate alerts every 24 hours. It will provide efficient monitoring and management of event logs through a web-based dashboard managed by the SOC or security administrator, with features such as a drop-down menu to select EventIDs. On selection, detailed information regarding the number of hosts affected by each of the selected EventIDs would be found. This is bound to improve the velocity and accuracy of threat detection and incident response, hence providing a system whereby organizations can rapidly fix security flaws, troubleshoot issues, and reduce the risk of exposure to cyber threats. Finally, this project should be a cost-effective enhancement of expensive SIEM solutions, boosting SMEs with limited financial resources for investments in cybersecurity with advanced event log analysis capabilities.

1.6 Report Layout

Chapter 1: Introduction

I have addressed the project Motivation, Rationale of the study, Related question and Expected output in this chapter.

Chapter 2: Background

I go over the project's specification and context. I have discussed the project's Related Works, Comparative Studies, Scope of the project and challenges in this chapter.

Chapter 3: Research Methodology

This chapter covers the Methods, Project working procedure, Project Workflow Diagram, Statistical Analysis and Implementation Requirements for the project.

Chapter 4: Experimental Result and Discussion

I have covered the Project Implementation, Experimental Result, and The Project Evaluation

Chapter 5: Summary, Conclusion, Recommendation and Implementation for Future work.

I have covered the Summary of the Project, Conclusion, Recommendation and Implication for the Further Study in this last chapter.

CHAPTER 2
BACKGROUND

2.1 Introduction:

It is essential to practice good cybersecurity measures in the digital era for any organization that wants to safeguard its IT assets from violations. Windows event logs remain one of the most important sources to monitor the system activities and look for security threats; however, analyzing the collected log information can be a complicated and time-consuming process that often results in errors. A significant number of organizations, particularly firms that are categorized as small to medium-sized enterprises (SMEs), discover that they are unable to absorb the cost of high-end security information and event management (SIEM) systems. To mitigate these problems, this project develops an "Automated Event Log Analysis and Alert System via Email" which sits on a Windows machine and makes use of Python scripting and Windows PowerShell to transmit log data, perform its analyses and disseminate the results. Another feature of it is that it consists of a centralized MySQL database in freesqldatabase.com and have friendly front end utilizing dashboard making it an efficient and affordable solution to improving cyber-security.

2.2 Related Works

1. The paper titled "System of Collection and Analysis Event Log from Sources under Control of Windows Operating System" by A.D. Moskvichev and M.V. Dolgachev addresses the inefficiencies of traditional event log analysis from Windows systems due to their increasing complexity. It proposes using Security Information and Event Management (SIEM) systems to centralize log collection and enable real-time analysis of security incidents. The research aims to develop a universal system for Windows event log collection and analysis, focusing on three methods: agent-based utilities, Windows Management Instrumentation (WMI), and Windows Event Forwarding (WEF). WEF, chosen for its low resource consumption and speed, was integrated with IBM QRadar SIEM using WinCollect. Testing demonstrated the system's effectiveness across various Windows versions. The study concludes that a WEF-based system significantly enhances

log management and security incident response, benefiting Security Operations Centers (SOCs) through centralized and efficient log analysis.

2. The paper titled “Automated Event Prioritization for Security Operation Center using Deep Learning” by Nitika Gupta , Issa Traore, and Paulo Magella de Faria Quinan deals with the inefficiencies in current Security Operation Centers brought on by the high volume and complexity of security data. The authors present a new approach to feature engineering by graphical analysis and subsequently employing a deep neural network to classify security events. The Security On-Demand dataset was preprocessed and engineered with newly identified features through graph-based visualization. A deep learning model was thus iteratively trained for scoring, achieving an AUC of 92.67% for marked improvements in classification accuracy. The research evidence suggests that this approach will help enhance the performance of a SOC, and future work will focus on further refining the feature engineering process and evaluating the efficiency of the approach through performance benchmarking.

3. The paper titled “A Strategy for Effective Alert Analysis at a Cyber Security Operations Center” by Rajesh Ganesan and Ankit Shah presents an optimization framework to enhance the performance of CSOCs by effectively managing alert data. The proposed strategy aims to maintain a high Level of Operational Effectiveness (LOE) through dynamic scheduling and optimal allocation of cybersecurity analysts. Key contributions include developing models for dynamic scheduling of both static and on-call analysts based on historical and predicted alert patterns, and an allocation model that optimizes analyst resources per shift. The LOE is measured using the total time for alert investigation (TTA), and the models are validated through simulation studies. This integrated approach provides a practical tool for CSOC managers to dynamically respond to varying alert demands and maintain operational efficiency.

2.3 Comparative Studies

The “Automated Event Log Analysis and Alert System via Email” project aligns with contemporary research emphasizing the need for automated and efficient log analysis systems. Moskvichev and Dolgachev's work on SIEM systems for centralized log analysis, Gupta et al.'s use of deep learning for event prioritization, and Ganesan and Shah's optimization framework for alert management collectively inform the project's approach to enhancing security event detection and response. By integrating automated log collection and analysis with a user-friendly dashboard for SOCs, this project offers a cost-effective solution tailored for small to medium-sized enterprises, improving their cybersecurity posture and operational efficiency.

The first study by Moskvichev and Dolgachev illustrates the benefits of a centralized system for log collection and analysis, which directly supports the project's aim to streamline and centralize log data for easier management and quicker response times. The use of Windows Event Forwarding (WEF) in their study showcases how a low-resource, efficient method can be implemented, aligning with the project's goal of minimizing resource usage while maximizing efficiency.

The second study by Gupta, Traore, and Quinan demonstrates how advanced machine learning techniques can prioritize and classify security events, emphasizing the importance of automated systems in handling large volumes of data. Although the “Automated Event Log Analysis and Alert System via Email” project does not currently employ deep learning, the concept of using automated tools to prioritize logs based on severity is a crucial component of the project, ensuring that critical events receive immediate attention.

The Third study by Ganesan and Shah highlights the significance of optimizing alert management and resource allocation in SOCs. Their findings on maintaining a high level of operational effectiveness through dynamic scheduling and optimal resource allocation are relevant to the project, which aims to improve incident response times and operational efficiency through automated log analysis and a user-friendly dashboard for SOCs.

2.4 Scope of the Project

The "Automated Event Log Analysis and Alert System via Email" project is designed to provide a comprehensive and accessible solution for the automatic collection, analysis, and distribution of data collected from Windows event logs. The scope of this project includes:

Automated log collection: This project will collect event logs from Windows systems according to user-defined parameters, such as time frames and e-mail addresses of recipients to whom reports must be sent. It will collect event log data from system, application, and security logs.

Log Analysis: The collected event logs will be analyzed for critical, warning, error, and AuditFailure events using Python scripting and Windows PowerShell commands. This analysis will help quickly identify possible security threats and system issues.

Data dissemination: It enables sending via email, storing on Google Drive, and a MySQL database for the safe and accessible management of log data.

Monitoring Dashboard: This web-based dashboard makes it easy for the SOC or the security administrator to monitor log data. The eventID selection drop-down menu will enable the detailed display of the number of hosts affected with respect to every eventID.

User Interaction: The system is provided with a user-friendly interface for the client to enter e-mail addresses, preferred periods for log collection, and select storage options of either email or Google Drive. Once the system configuration is in place, it will run every 24 hours by Task Scheduler without user interaction.

Cost-Effective Solution: This project, using open-source tools and technologies, provides SMEs with a cost-effective solution to expensive SIEM solutions for implementing advanced event log analyses.

Enhanced Security Posture: Automated systems improve threat detection and response, enabling overall cybersecurity and improved system monitoring capabilities for the IT environment.

Limitations: There is no feature in the project about identification and automatic handling of the necessary issues; it is designed only for detection and sending alert; handling of detected issues can be done by the SOC or security administrator only. Moreover, the system was also not designed with the integration in machine learning systems that are used in advanced threat identification, and predictive analysis.

2.5 Challenges

This "Automated Event Log Analysis and Alert System via Email" project is quite innovative and helpful. However, for its successful implementation and running, various challenges are to be addressed.

Data privacy and security: No sensitive log data should be compromised during transmission or storage. This will ensure a robust encryption method to safely send the data in email and store it on Google Drive and MySQL databases.

Integration and Compatibility: Integration into most Windows environments, its compatibility with the large number of Windows versions is still a huge challenge. Quite often, the system shall require configuration adjustments or updates to the same effect of keeping the whole system in line.

Handling Large Volumes of Data: Since the system is supposed to collect and analyze log data after periodic intervals, it has to handle large amounts of data without degradation in performance. Efficient techniques in managing and optimizing data are essential to prevent system overload.

Performance Impact: It should be ensured that the system collects and analyzes logs in a manner that does not grossly impact the performance of the systems to be monitored. If

this line cannot be drawn, general or specific user dissatisfaction may arise due to degradation in system performance brought about by high resource usage.

User Adoption and Training: Proper training and support should be provided so that the system is effectively workable by either SOC or security administrators. The UI and dashboard should be user-friendly to ensure easy adoption and usage.

False Positives and Noise: An automated log analysis system may either create a noise of false positives or overwhelm non-critical events. Therefore, fine-tuning such a system to classify and prioritize different events by their severity correctly is essential to avoid cases of alert fatigue and take timely action on critical ones.

Resource Constraints: This is especially common in SMEs, which typically have limited financial and technical resources. Ensuring that the system will remain cost-effective and not require extensive resources for its implementation and operation is crucial to the diffusion and sustainability of this system.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter details the methodology adopted for the "Automated Event Log Analysis and Alert System via Email" project. This includes the project subject and instruments, methods of data collection and analysis, working procedure, statistical analysis techniques, and implementation requirements. The methodology is targeted at providing a structured approach to the development and deployment of an automated system in event log analysis, targeting efficiency, accuracy, and cost-effectiveness in the monitoring of IT environments especially within small to medium enterprises (SMEs).

3.2 Project Subject and Instruments

Subject:

The main focus of this project is in automating event log analysis, an essential part of cybersecurity. The Windows event logs, which include the Application, System, and Security logs, are important in monitoring system activities. The project addresses the issue of the cost-effective, efficient way for SMEs in collecting, analyzing, and taking actions upon their log data with no need for expensive SIEM solutions.

Instruments:

A combination of software tools and technologies were used to achieve the project objectives:

PyCharm: An IDE intended for writing and debugging Python scripts that make up the core of a project.

Python: This is the primary scripting language used in automating the collection, analysis, and dissemination of event log data.

Python Tkinter: A standard Python library for GUIs, which provides the user interface to input options—like email addresses, time frames, and storage selection options (Email or Drive) for the executable file that will be presented to the client.

Python smtplib: This module is used to send emails with the collected log data within CSV files.

Windows PowerShell: Used in Python scripts to run commands to collect log data from Windows environments.

Freesqldatabase.com: Comes with a web-based service hosting MySQL databases for log data storage and management.

phpmyadmin.co: An open-source web-based application to handle MySQL databases; it will store and pull log data for display on the dashboard.

PHP: Used to collect data of interest from the MySQL database and also display this on the dashboard.

XAMPP Server: It is used for hosting the dashboard, acting as an intercross-platform web server. This shall allow local monitoring of log data.

Task Scheduler: The Windows utility is designed to automate the periodic execution of an EXE file, thus ensuring the continuous collection of log data without any form of user intervention.

Google Drive: Used as an additional storage option for log data.

3.3 Methods

The methods implemented in this project entail a couple of steps designed to automate Windows event log data collection, analysis, and dissemination. These significant steps are instrumental in ensuring that the system works efficiently and effectively concerning system monitoring and incident response.

3.3.1 Automated Event Log Data Collection:

Python Script and PowerShell Integration: Python script is integrated with Windows PowerShell commands to collect event logs from Windows systems. This script converted to a Windows executable file for clients to install.

User Input Interface: Using Python Tkinter, design a GUI requiring the user to input an email address and a duration in hours to collect logs.

3.3.2 Automated data Transmission:

Email Alerts: Utilizing the smtplib module in Python, it emails data gathered on logs in CSV files to an email address specified.

Task Scheduling: Used Windows Task Scheduler to allow the EXE to run every 24 hours to enable continuous and automatic log collection and e-mail alert notification without human intervention.

3.3.3 Centralized data storage and dashboard monitoring:

MySQL Database and Google Drive Integration: Log data is stored through a centralized MySQL database hosted on phpmyadmin.co site and, optionally, on Google drive for redundancy.

Web-Based Dashboard: This dashboard has been developed using PHP; hereby an SOC or any Security Administrator will be allowed to monitor log data and select events from a drop-down menu to view details of affected hosts for this selected EventID. This dashboard hosted on an XAMPP server for local access.

3.4 Project Working Procedure

"Automated Event Log Analysis and Alert System via Email" follows a structured working procedure that effectively collects, analyses, and disseminates event log data from Windows. This procedure is user-friendly and does not have many interactions, but it elicits important insights and notifications from the system events. Below are the major steps involved in this procedures:

1. Conversion of Python Script to EXE:

This Python script, which integrates Windows PowerShell commands for the collection of log data, is converted into an executable (EXE) file to install on endpoint clients.

2. User Input Interface:

When the clients run the installation file first time in 'run as administrator' mode, a user interface created with Python Tkinter is shown.

The UI needs three different inputs from the user:

- **Email:** Specifies the recipient for the log data collected.
- **Hours:** This indicates the number of hours of logs that need to be collected from the Windows Event Viewer.
- **Options:** This enables the user to either send log data via email, or store it on Google Drive.

3. Log Data Collection:

- Once the 'Get-Log' button is clicked, it gathers log data based on the provided parameters. This data is collected from the Windows System, Applications, and Security logs. It will focus on Critical, Warning, Error, and audit failure events.

4. Data Transmission and Storage:

- Log data collected is formatted into CSV files.
- Depending on the selected option the system sends these CSV files to the specified email address using the smtplib module or the log data is stored on Google Drive.
- Moreover, this log data is stored in a MySQL database hosted on phpmyadmin.co site.

5. Scheduling of Tasks:

- The user inputs email, hours and selected option (Email or Drive), are stored for future user.
- The exe runs every 24 hours in the Windows Task Scheduler. Such automation will ensure, quite nearly, that log collecting, forwards, and storage take place periodically without manual intervention.

6. Dashboard Integration:

- A web-based dashboard is developed using PHP language and hosted on an XAMPP server.
- It allows the SOC or Security Administrator to monitor event log data centrally from the dashboard.

- On a drop-down menu, administrators can select the EventIDs, to view with detailed information on exactly how many hosts are affected by that particular EventID.
- It gives a dynamic update of the dashboard, through which one can know the status of the IT environment in any organization.

7. Data Filtering and Display:

- Filter out relevant log data from a MySQL database and display it on the dashboard by using PHP scripts.
- The system ensures that only significant events (critical, warning, error, AuditFailure) are highlighted, enabling efficient threat detection and response.

3.4.1 Project Workflow Diagram

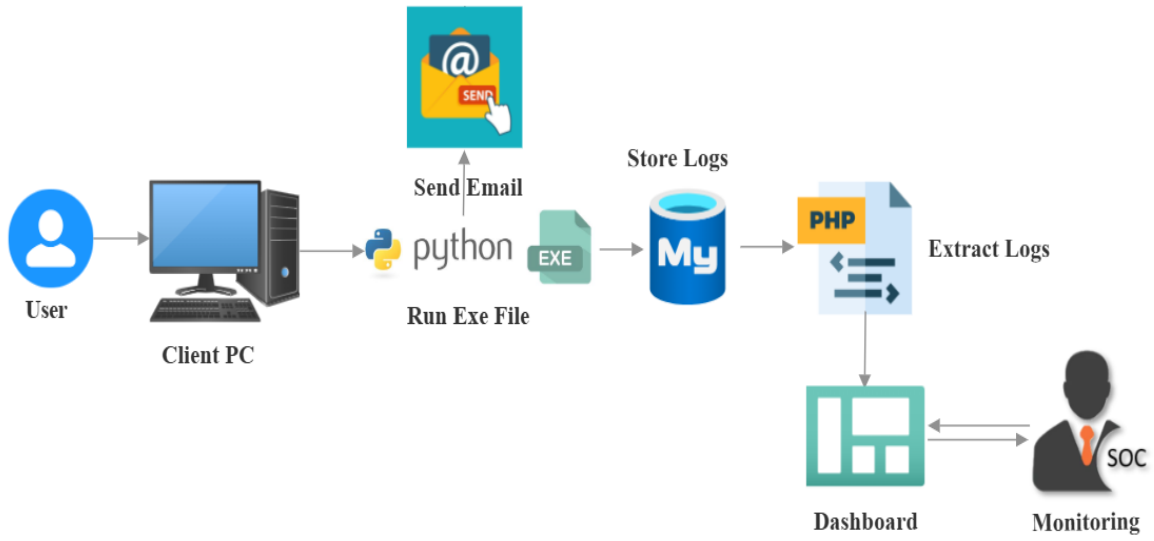


Figure: 3.4.1: Project Workflow Diagram

3.5 Statistical Analysis

The “Automated Event Log Analysis and Alert System via Email” project applies statistical analysis to mining essential information from the pooled event log data. It will allow stakeholders at large to make informed decisions and prioritize mitigation efforts by applying statistical techniques to identify patterns, trends, and anomalies within the log data. Key Statistical Methods:

Frequency Analysis: Frequency analysis is using the tool to establish how often a particular type of event, for example, critical, warning, error, and AuditFailure events, occurs from the logs collected.

Severity Analysis: For performing the severity analysis, events can be categorized into low, medium, and high severity.

3.6 Implementation Requirements

The “Automated Event Log Analysis and Alert System via Email” project will require the implementation of specific specified software, hardware, and network resources to be functional. The detailed requirements are given below:

3.6.1 Software and Tools Requirements:

Python: This will be the primary programming language for scripting all automation processes.

PyInstaller: Used for packaging Python scripts into an executable format, such as EXE, which is self-contained and can then be deployed on the client's machines.

Tkinter: This is Python's de facto library for developing GUIs.

smtplib: This is a Python package for sending mail via the Simple Mail Transfer Protocol. It allows an to send email alerts.

Windows PowerShell: It has been integrated into Python scripts to retrieve log data from Windows systems.

MySQL Database: This holds at phpmyadmin.co site, keeping a central storage location for log data collected.

PHP: Used to create scripts for extracting log data from the MySQL database and displaying it on the dashboard.

XAMPP Server: Used to host web-based dashboard to locally access.

Google Drive API: This was used to store log information on Google Drive.

3.6.2. Hardware Requirements:

Client Machines: These are the endpoints on which the EXE file will be executed. Performances of such machines should and full of storage space enough to sustain log data collection and transmission.

Dashboard for Server Hosting: A server or Windows system needs to run the XAMPP server to host the web-based dashboard.

3.6.3 Network Requirements

Stable Internet Connection: Required for sending emails, accessing the MySQL database.

3.6.4. User Requirements:

Administrative Privileges: Clients should run the EXE file with 'run as administrator' rights so that the script has access to all necessary logs on the system to perform the specified actions.

Input Parameters: The users need to provide input such as email, Hours, and options. These inputs will be stored for future use, allowing the system to run automatically without further user interaction.

3.6.5 Additional Tools and Resources:

Google Drive: This will be used as optional storage for log data and data redundancy for easy access.

Task Scheduler: This processes the EXE file every 24 hours, making the collection, transmission, and storage of logs automatic without manual intervention.

CHAPTER 4
EXPERIMENTAL RESULT

4.1 Introduction

The chapter "Experimental Result and Discussion" discusses the implementation results of the project in detail and analyzes the empirically achieved results. This chapter will explain to the reader the experimental process and the methodologies used and explain the significance of the results achieved for achieving the objectives of the project. This chapter analyzes the experimental data in a methodical way, which provides readers with a deeper insight into the performance, functionality, and effectiveness of the automated event log analysis system. Stakeholders will, in turn, learn more about the system in terms of performance, capabilities, limitations, and possible areas for improvement following an in-depth analysis of the experimental results.

4.2 Project Implementation

User Interface: For the first execution, the user needs to provide input such as email, hours, and the selected option (email or drive). Then these inputs will be stored for future use. Then, using this information, it will automatically execute every 24 hours based on the Task Scheduler without user interaction.

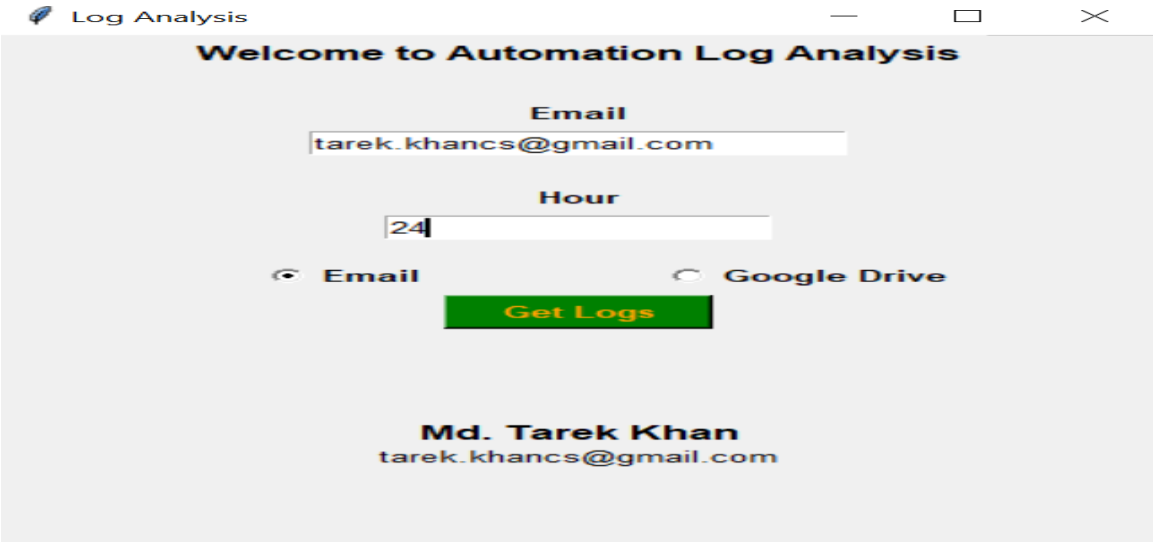


Figure 4.2.1: User Interface for sends email

IDE: This project was written in the Python programming language with PyCharm to process, analyze logs, and generate outputs corresponding to Application, System, and Security logs. These outputs were then mailed to the SOC or Security admin email and saved in a MySQL database and google drive.

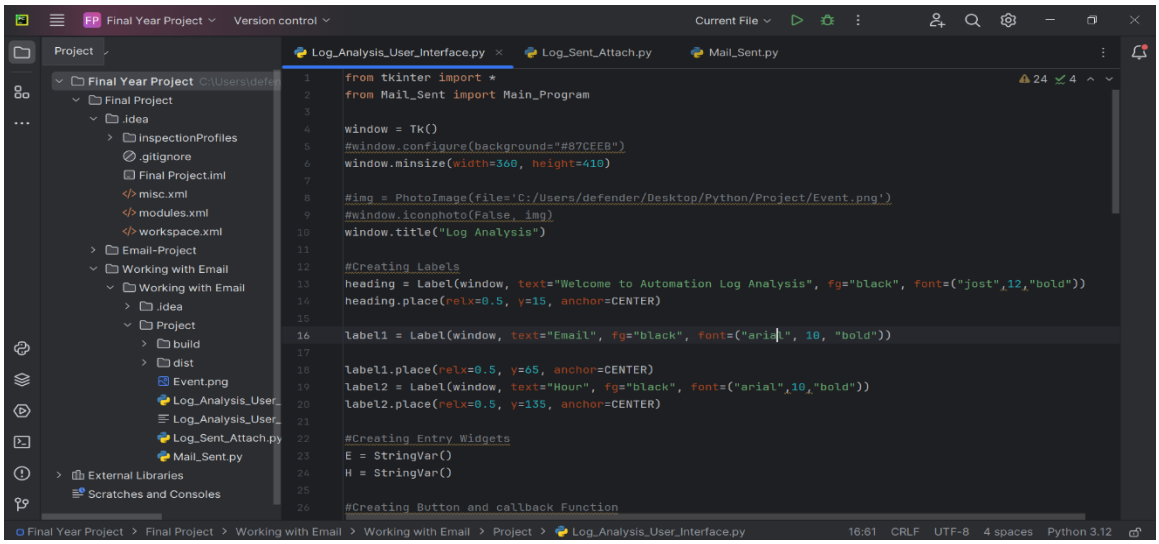


Figure 4.2.2: IDE Code Implementation Part

MySQL Database: This MySQL database component is the store of all processed data after the implementation phase, and it stores only the Error, Warning, and Critical data to have in-depth analysis.

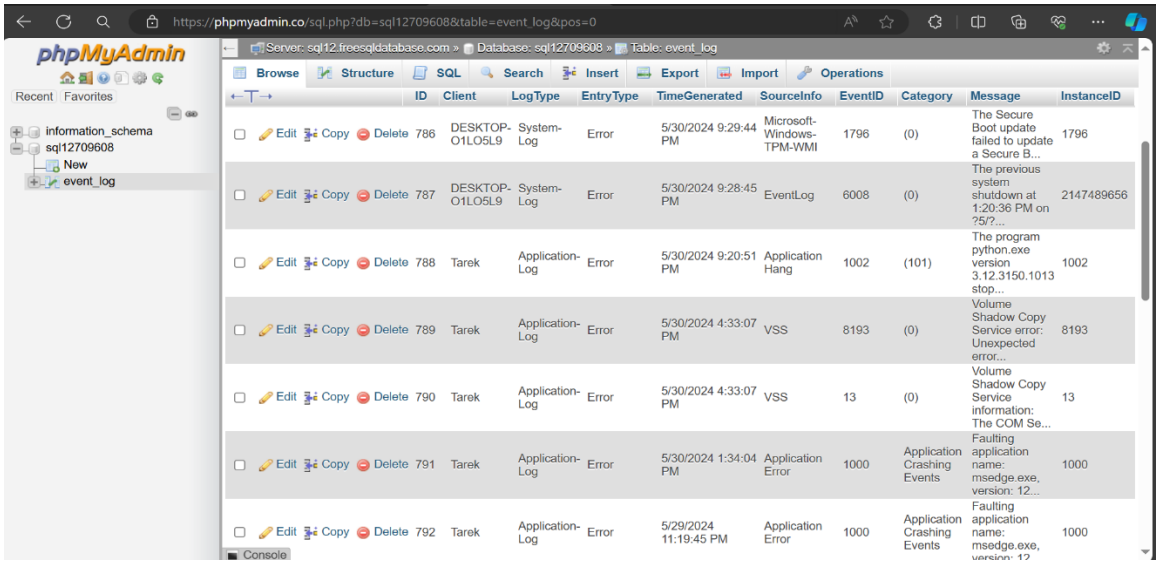


Figure 4.2.3: Data Storage Part on MySQL Database

4.3 Experimental Result

Email data: All processing is reported to the SOC or Security Admin email.

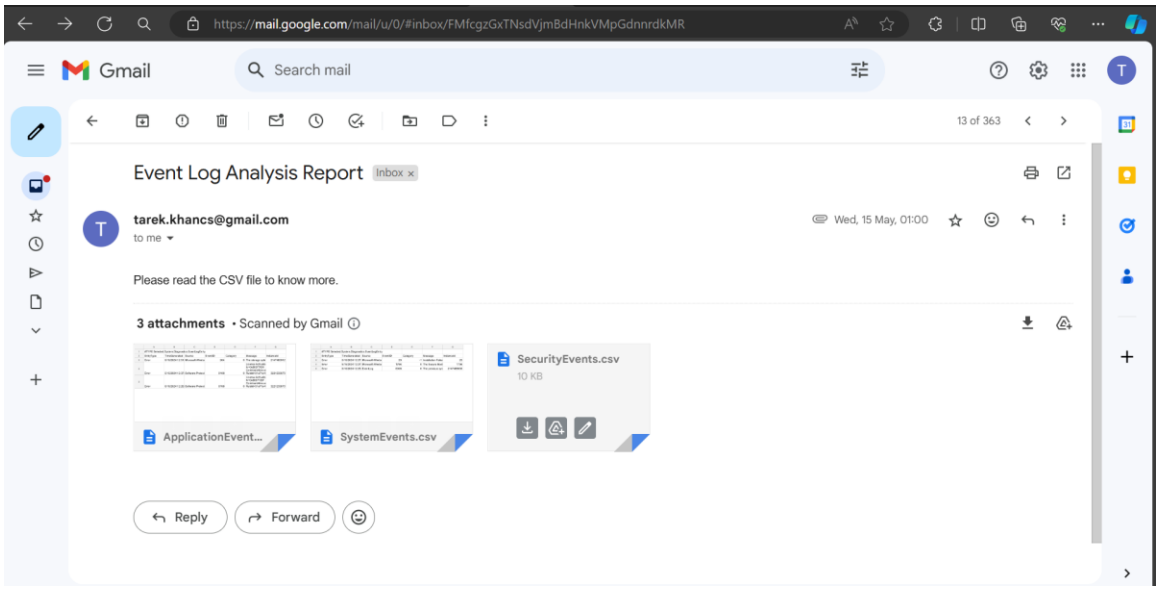
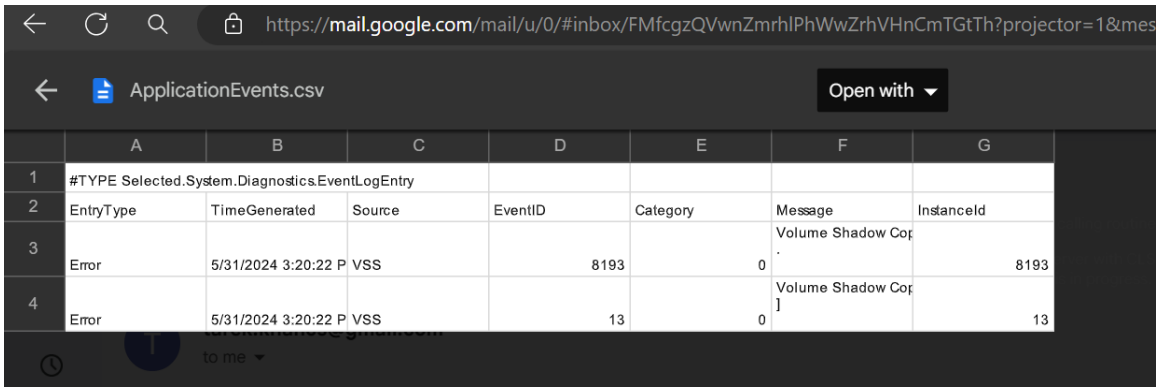


Figure 4.3.1: Event Log through Email

Applications Logs in email: This part is the collection of the Application Event log data in the form of a CSV file.



	A	B	C	D	E	F	G
1	#TYPE Selected.System.Diagnostics.EventLogEntry						
2	EntryType	TimeGenerated	Source	EventID	Category	Message	InstanceId
3	Error	5/31/2024 3:20:22 P	VSS	8193	0	Volume Shadow Cop .	8193
4	Error	5/31/2024 3:20:22 P	VSS	13	0	Volume Shadow Cop]	13

Figure 4.3.2: Application Event Log

System Logs in email: This part is the collection of the System Event log data in the form of a CSV file.

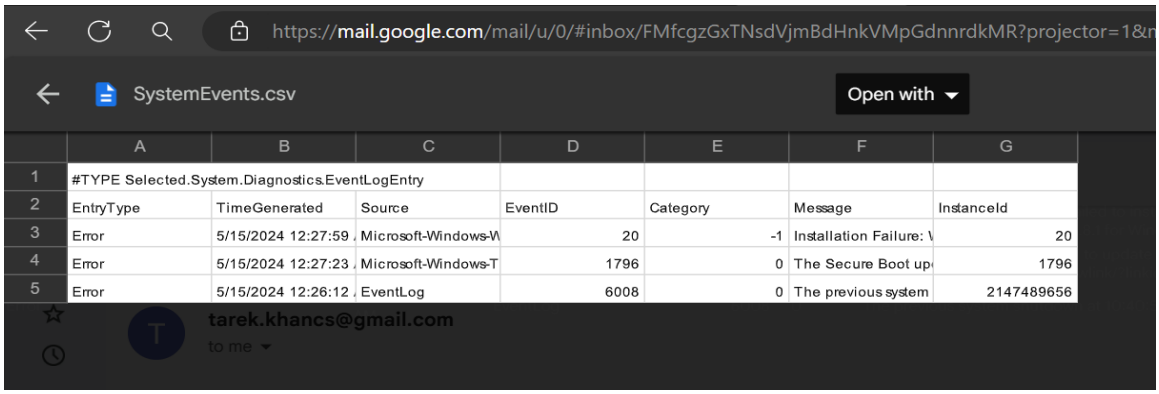


Figure 4.3.3: System Event Log

Security Logs in email: This part is the collection of the Security Event log data in the form of a CSV file.

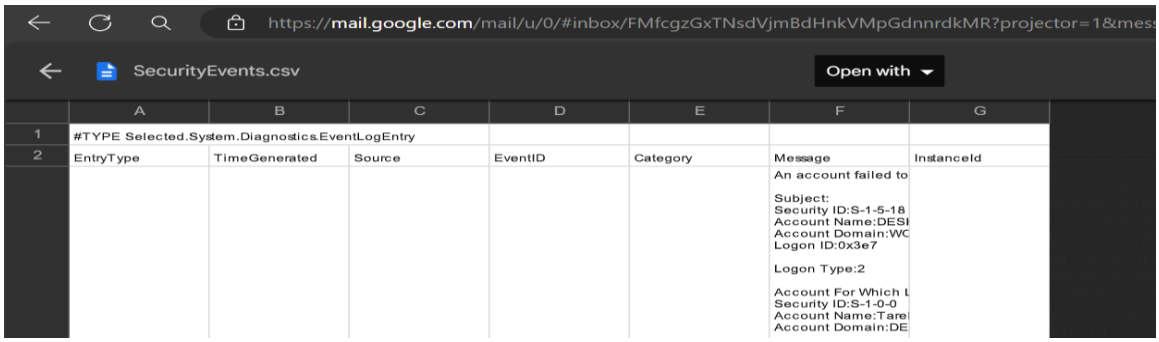


Figure 4.3.4: Security Event Log Part-1

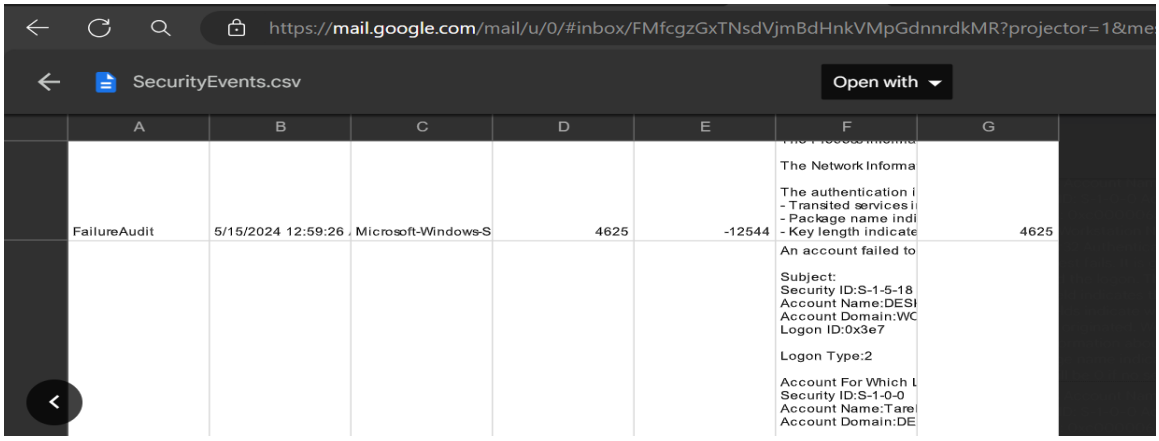


Figure 4.3.5: Security Event Log Part-2

Stored on Google Drive: This is an optional storage for log data and data redundancy for easy access.

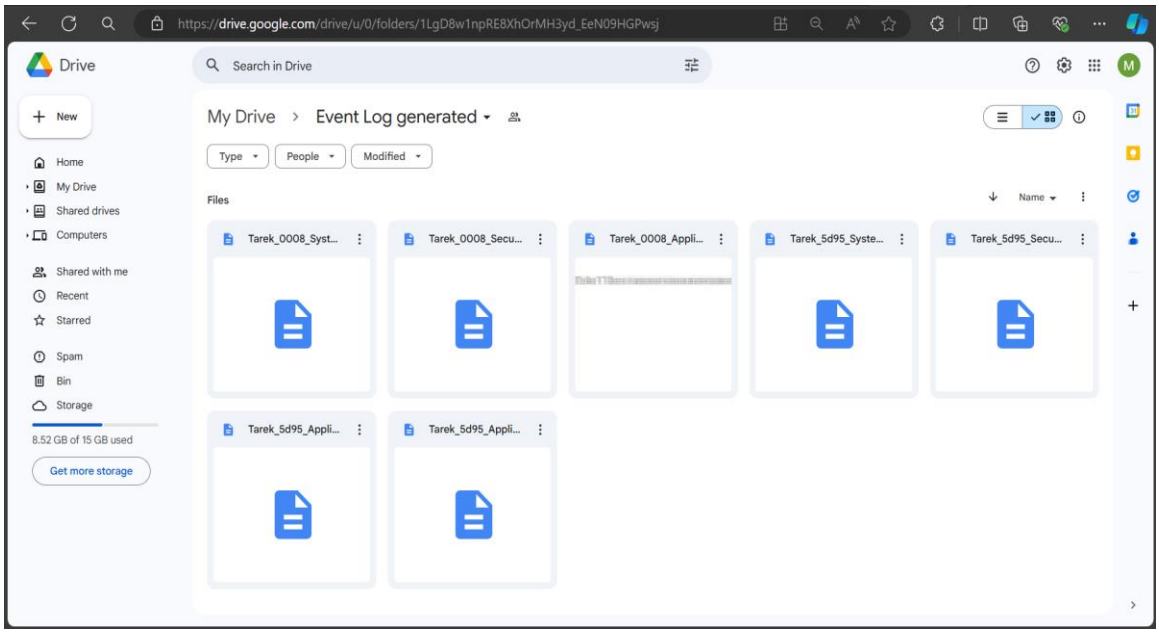
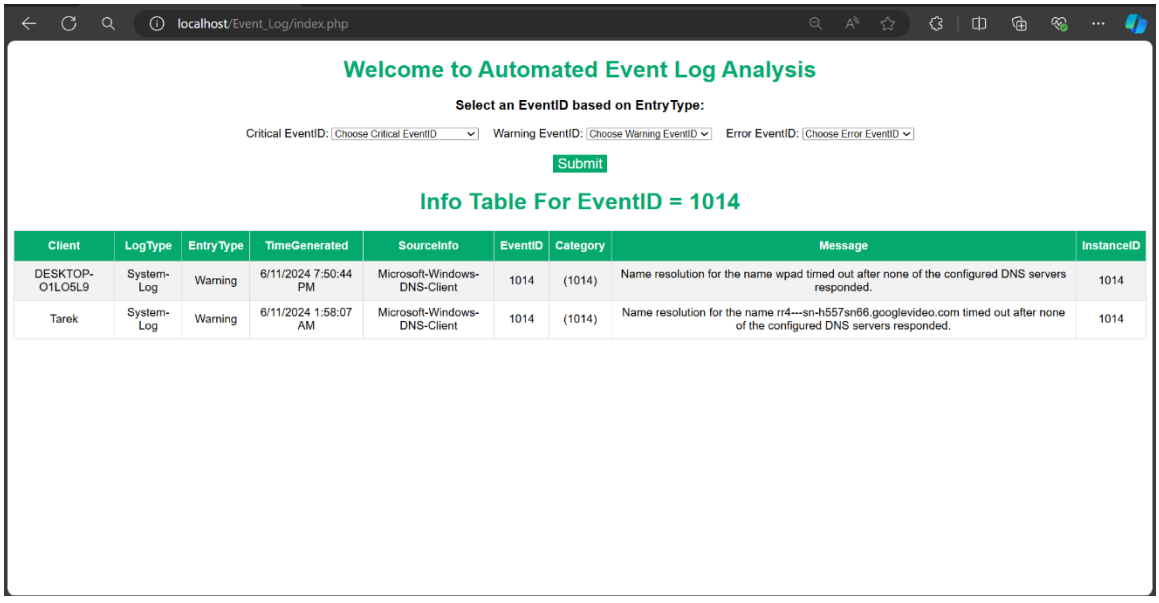


Figure 4.3.6: Stored on google drive

SOC or Security Admin Dashboard: This is the Dashboard section where the SOC or Security admin is able to view and gather information on infected devices. They need to choose an EventID after which the output will give a list of infected Hosts.



Client	LogType	EntryType	TimeGenerated	SourceInfo	EventID	Category	Message	InstanceID
DESKTOP-O1LO5L9	System-Log	Warning	6/11/2024 7:50:44 PM	Microsoft-Windows-DNS-Client	1014	(1014)	Name resolution for the name wpad timed out after none of the configured DNS servers responded.	1014
Tarek	System-Log	Warning	6/11/2024 1:58:07 AM	Microsoft-Windows-DNS-Client	1014	(1014)	Name resolution for the name rr4---sn-h557sn66.googlevideo.com timed out after none of the configured DNS servers responded.	1014

Figure 4.3.7: SOC Analyst Dashboard

4.4 The Project Evaluation

The evaluation of the project "Automated Event Log Analysis and Alert System via Email" focused on key areas:

Monitoring Accuracy: It was accurate at monitoring Windows event logs and neatly categorized events into critical, warning, error, and audit failure. This information on the dashboard was available in detail regarding affected hosts for enhanced threat identification and response.

Performance Impact: By running the EXE file every 24 hours using Windows Task Scheduler, this system ensured that performance impact on host machines was kept to a minimum, hence ensuring effective data collection and transmission with very minimal resource consumption.

Data Security and Privacy: During transportation and storage, log data was enveloped in robust encryption methods. Access to the dashboard was strictly limited to authorized personnel to protect the data from any unauthorized disclosure and to protect its integrity.

User Feedback: Users, SOC, and Security Administrators have especially liked the user-friendly interface and automated functionality.

Cost-effectiveness: It offered an affordable alternative to the traditional SIEM solution with open-source tools. It offered leading-edge log analysis without significant investments and became very accessible to SMEs.

Continuous Improvement: The project focused on constant upgrading and improvement concerning feedback from users and new cyber threats. The system is adequate and relevant through ongoing monitoring.

CHAPTER 5
SUMMARY, CONCLUSION, RECOMMENDATION, AND IMPLICATION FOR FUTURE IMPROVEMENT

5.1 Summary of the Project

The “Automated Event Log Analysis and Alert System via Email” project was designed to overcome the shortcomings and challenges of such a manual procedure in log analysis in IT environments. It was to be done with a low-cost, automated system enabling SMEs to monitor and act upon security events without the presence and expense of an SIEM system. Automation of data collection, analysis, and dissemination of the Windows event log data is done through Python as the scripting language alongside Windows PowerShell commands in this project. This system starts and thereby collects event logs based on pre-set email addresses as well as filters of the desired timeframe and then classifies them in terms of critical, warning, error, as well as AuditFailure events. The log data is then sent in email and Google-Drive as CSV files and centrally stored in a MySQL database hosted in phpmyadmin.co site.

The system to the capability of a critical feature, once set for the first time, operates 'autonomously.' Once these inputs - email, hours are provided by the client along with the choice of storage, either email or Google Drive; the system shall run the EXE file using Task Scheduler every 24 hrs. This shall continuously collect the data and monitor it without any further interaction with the user. A web-based dashboard is also available, which is run and administered by the Security Operation Center, SOC, or Security Administrator. The Eg: dashboard with a drop-down menu selection of any specific EventIDs the dedicated data detailed member of the number of hosts getting affected by that particular EventID. This will allow SOC teams to raise alerts or responses to those events corresponding to the severity of the respective event.

In a nutshell, this is all about how the project “Automated Event Log Analysis and Alert System via Email” would aid in accelerating threat detection and incident response with

accuracy. It offers such a user-friendly automated event log analysis solution to the end that it empowers SMEs to keep their cybersecurity and operational efficiency at a very high level, improving their overall security posture and helping companies protect and troubleshoot their computer systems in IT environments.

5.2 Conclusion

The project solution to the "Automated Event Log Analysis and Alert System via Email" project is timely and significant for the need to analyze the logs effectively and economically across diverse IT environments, especially SMEs. This project identifies the shortcomings of the traditional methods and, as such, sought to collect, automatically analyze, and delineate entries. The result is a very robust solution scalable to a high degree because many open-source tools are integrated with the Python scripting, and Windows PowerShell commands are incorporated to do the work. In-place system capability detects security events and the user interface and automates the various operations done, which brings much speed and accuracy to threat detection, troubleshoot issues and incident response. The web-based dashboard offers all these insights in real-time so that the SOCs or Security Administrators can make informed decisions and get the priorities for response right by event severity. Eventually, this project would help improve Organizations' overall cyber security posture and operational efficiency so that the organizations always get to monitor and proactively secure their IT environment from ever-evolving threats.

5.3 Recommendation

1. Continuous Monitoring and Improvement: One may recommend continuous monitoring of the automated event log analysis system regarding performance and effectiveness. There will be systematic reviews in relation to threat detection accuracy, incident response times, and also other problems or challenges experienced during operations. This way, improvements and updates can be made in enhancing system performance or reliability.

2. Integration of Advanced Analytics: This involves the integration of advanced analytics to include machine learning algorithms to extend the functionality of the system towards

threat detection by way of anomaly detection. The machine learning algorithm will allow for trends and patterns to be established in log data that can assist in identifying possible security threats and anomalies.

3. Integration with Other Security Tools: The automated event log analysis system, when combined with other security-oriented tools and technologies, would broaden the threat-hunting features beyond the incident response phase. It would be exciting to combine the automation of current event log analysis systems with self-sufficient intrusion detection systems like IDS antivirus Software or network monitoring software to develop a fully-fledged security tool.

4. Regular Backups and Planning on Disaster Recovery: Implement comprehensive backup and disaster recovery plans that will enable the integrity and availability of log data kept in the central MySQL database and on an independent platform such as Google Drive. All log data will be backed up at periodic stages while practicing various disaster recovery procedures to minimize data or system downtime loss due to unexpected incidents or failures.

5. Feedback and Collaboration: Elicit cooperation and feedback from the SOC analysts, security administrators, and other stakeholders using this system. Take their opinions about the performance of the system, its ease of use, and the improvement needed in its features. Quite obviously, against this backdrop, their suggestions would pertain to improvisation and priority for the development team in the times ahead. This will be the essence of a collaborative environment where the system would reflect the dynamic needs and requirements of the organization.

5.4 Implication for Further Study

Implications of further study in the scope of event log analysis automation and cybersecurity incident response could include diving deeper into specific areas for refinement and expansion. One of the first implications would be to conduct more research on the application of machine learning algorithms for advanced log analysis and anomaly detection to improve the system. Investigate the possibility of integrating more data sources into the analysis of Windows event data, for instance, network logs, server logs, and application logs, to paint a more complete picture of cybersecurity incidents. Studying the feasibility of real-time response and monitoring mechanisms would assist an organization in preemption of immediate security threats. Moreover, organizational effectiveness in different organizational contexts, such as specific industry use cases or different network architectures, can be evaluated to derive the implementation and best practices.

REFERENCES

[1] N. Gupta, I. Traore, and P. M. F. de Quinan, "Automated event prioritization for security operation center using deep learning," in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 5864-5872.

[2] A. D. Moskvichev and M. V. Dolgachev, "System of Collection and Analysis Event Log from Sources under Control of Windows Operating System," in *2020 IEEE Far East Con International Conference on Electrical Engineering, Electronics and Instrumentation (FarEastCon)*, 2020, pp. 1-5. DOI: 10.1109/FarEastCon50210.2020.9271520.

[3] R. Ganesan and A. Shah, "A strategy for effective alert analysis at a cyber security operations center," in From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday, 2018, pp. 206-226.

[4] R. Ranjan, "Windows Event Log Analysis & Incident Response Guide," Medium. [Online]. Available: https://medium.com/@rajeevranjancom/windows-event-log-analysis-incident-response-guide-739af79b518b. [Accessed: Apr. 20, 2024].

[5] Microsoft, "Collect Windows event log data sources with Log Analytics agent," Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/azure/azure-monitor/agents/data-sources-windows-events. [Accessed: May 15, 2024].

[6] W3Schools, "Learn about MySQL," W3Schools. [Online]. Available: https://www.w3schools.com/MySQL/default.asp. [Accessed: Apr. 03, 2024].

[7] Javatpoint, "Learn about Python," Javatpoint. [Online]. Available: https://www.javatpoint.com/python-tutorial. [Accessed: Feb. 15, 2024].

[8] Microsoft, "Learn about Windows PowerShell," Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/training/paths/get-started-windows-powershell/. [Accessed: Feb. 29, 2024].

[9] Python Software Foundation, "Learn about Python Tkinter," Python.org. [Online]. Available: https://docs.python.org/3/library/tkinter.html. [Accessed: Mar. 06, 2024].

[10] Python Software Foundation, "Learn about smtplib," Python.org. [Online]. Available: https://docs.python.org/3/library/smtplib.html. [Accessed: Apr. 16, 2024].

Automated Event Log Analysis And Alert System Via Email.pdf

ORIGINALITY REPORT

15%	15%	4%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	6%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	4%
3	Submitted to University of Queensland Student Paper	<1%
4	researchberg.com Internet Source	<1%
5	link.springer.com Internet Source	<1%
6	epub.uni-regensburg.de Internet Source	<1%
7	A.D. Moskvichev, M.V. Dolgachev. "System of Collection and Analysis Event Log from Sources under Control of Windows Operating System", 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), 2020 Publication	<1%