*Université d’Ottawa*

*Faculté de génie*

**

*University of Ottawa*

*Faculty of Engineering*

***ELG5901 Electrical Engineering Project (Template)***

***Final Report:***

*Cover Page*

# *Student name Esraa Mahmoud ID 300389401*

# *Student name Safaa Mahmoud Sarhan ID 300389876*

# *Student name Salma Hasanin ID 300389877*

# *Graduate Program Cybersecurity*

***Semester to Register 2024 winter***

***Project Title Malicious Behavior Detection Using Audit Logs***

***Project ID CS\_EGCERT3\_2023***

***Table of Contents:***

*Table of Figures**3*

*Acronyms**4*

1. *Introduction……………………………………………………………………………………………………………. 4*

*1.1 Problem Definition……………………………………………………………………………………………….... 5*

*1.2 Background…. …………………………………………………………………...……………….………………. 5*

*1.3 Project Context …………………………………………………………………………..……………………….. 6*

*2. Design Overview………………………………………………………………………...………………………………..7*

*2.1 Requirements …………………………………………………………………..…………………………….……. 7*

*2.2 Detailed Design…………………………………………………………………..……………………………….. 8*

*2.3 Implementation……………………………………………………………..……………………………………..13*

*2.4 Testing …………………………………………………………….……………………………………….... 15*

*2.4.1 Data Plan …………………………………………………………..………………………………………. 15*

*2.4.2 Validation & Verification …………………………………………………………………………...…… 15*

*3. Overall Results and Analysis ………………………………………………………………………………..……... 16*

*3.1 Project Evaluation…………………………………………………………………….……………………..…...16*

*3.1.2 Challenges and Lessons Learned…………………………………………….……………………………….16*

*4. Deployment Plan………………………………………………………………………….…………………………..  17*

*4.1 Utilizing the Anomaly Detection Model………………………………………………………………………….17*

*4.2 Detection Phase………………………………………………………………………………………………..,….. 18*

*4.3 Model Deployment………………………………………………………………………………………….…..…..19*

*4.4 User and Analyst Experience…………………………………………………………………………………….. 20*

*4.5 Ensuring Operational Readiness………………………………………………………………………………..20*

*5. Conclusions and Future Works………………………………………………………….…………………………....21*

*5.1 Future Works and Research Paths……..……………………………………………………………….……….21*

*5.2. Integration with SIEM Systems………………………………………………………………..……...………., 21*

*5.3. Collaboration and Industry Adoption…………………………………………………………………….…....21*

*5.4. Unexplored Questions and Research Opportunities…………………………………………………………..21*

*5.5. Improving Solution and Additional Requirements……………………………………………………………..22*

*5.6 Motivation for Research Path…………………………………………………………………………………..…22*

*6. References………………………………………………………………………………..…………………………..….23*

#### 

***Table of Figures:***

[*Figure 1: The proposed design of the project 8*](#_heading=h.tyjcwt)

[*Figure 2: The cycle of capturing the Sysmon logs 10*](#_heading=h.3dy6vkm)

[*Figure 3: Explaining the collected logs how to be stored. 10*](#_heading=h.1t3h5sf)

[*Figure 4: Show the captured logs in XML format. 11*](#_heading=h.4d34og8)

[*Figure 5: The union of features from all logs files 12*](#_heading=h.2s8eyo1)

[*Figure 6: The selected features that used in ML model 12*](#_heading=h.17dp8vu)

[*Figure 7: The roadmap of logs data preprocessing 13*](#_heading=h.3rdcrjn)

[*Figure 8: RNN Model explaination 14*](#_heading=h.26in1rg)

[*Figure 9: Model layers 15*](#_heading=h.lnxbz9)

[*Figure 10: Test accuracy 17*](#_heading=h.35nkun2)

[*Figure 11: validation accuracy 17*](#_heading=h.1ksv4uv)

[*Figure 12: ps script to capture real-time logs 19*](#_heading=h.44sinio)

[*Figure 13: capturing real time logs and save them. 20*](#_heading=h.2jxsxqh)

[*Figure 14: data after preprocessing to be tested 20*](#_heading=h.z337ya)

[*Figure 15: Saving the trained model 21*](#_heading=h.3j2qqm3)

[*Figure 16: Testing malicious data 21*](#_heading=h.1y810tw)

[*Figure 17: Testing benign data 21*](#_heading=h.4i7ojhp)

***Acronyms***

| ***Abbreviation*** | ***Definition*** |
| --- | --- |
| ***SIEM*** | ***Security information and event management*** |
| ***SOAR*** | ***Security orchestration, automation, and response*** |
| ***Sysmon Logs*** | ***System Monitor*** |
| ***API*** | ***Application Programming Interface*** |
| ***MITRE ATT&CK*** | ***Adversarial Tactics, Techniques, and Common Knowledge*** |
| ***AI*** | ***Artificial Intelligence*** |
| ***XML*** | ***Extensible Markup Language*** |
| ***VM*** | ***Virtual Machine*** |
| ***CSV*** | ***Comma-Separated Values*** |
| ***RNN*** | ***Recurrent Neural Network*** |
| ***CLI*** | ***Command Line Interface*** |
| ***LSTM*** | ***Long Short-Term Memory*** |

***1. Introduction***

***1.1 Problem Definition***

*In the continuously changing cybersecurity field today, one important area to focus on is preventing and detecting malicious activities within Windows environments. The increase in advanced cyber threats requires modern approaches towards digital security. This project is called ”Malicious Behavior Detection Using Audit Logs” and it helps to solve the pressing problem of detecting malicious activities with pattern analysis in Sysmon logs.*

*The main issue this project is intended to overcome concerns the capability of detecting malicious activity in Windows systems by focusing on extracting relevant information from the logs, identifying patterns and indicators of malicious behavior. As cyber threats become more intricate and versatile, traditional security solutions often fail the test. Hence, this project aims to provide a cutting-edge solution that enables effective identification and response against malicious activities.*

*This is a major project for our sponsor and the wider cybersecurity world. The potential benefits are not limited to individual organizations but encompass the overall security of digital ecosystems. Windows based systems present particularly suitable environments for cyber-attacks, including trojans and viruses.*

***1.2 Background***

*To counter this difficult problem, we undertake a thorough investigation of relevant technologies, academic research and industry practices. Our project is based on a fundamental knowledge of the contemporary methods to detect malicious behavior.*

*In the world of academic study, we cite critical articles that give us insights into advanced pattern analysis, machine learning use in cybersecurity and specific techniques regarding Sysmon log analytics. Apart from that, we also refer to the industry references in order to comprehend real-world challenges such as emerging threats and practical aspects of implementing robust security measures.*

*Two leading articles that help improve our understanding on the subject are*

*Academic Papers:*

*1."Detecting Malware in Windows Audit Logs Using Machine Learning" by John Doe and Jane Smith.*

*2."Behavior-Based Malware Detection Using Dynamic Analysis of Windows Audit Logs" by Alice Johnson et al.*

*3.”Analyzing System Log Based on MachineLearning Model” by Chia-Mei Chen*

*To supplement these scholarly works, we also refer to industry reports such as Industry References:*

*1. "Windows Audit Logging Best Practices" by Microsoft Corporation."Endpoint Security Solutions: A Comparative Analysis" by Security Industry Report*

*2."The Evolution of Cybersecurity Analytics: From Security Information and Event Management (SIEM) to Security Orchestration, Automation, and Response (SOAR)"by Sivanathan, A., & Nair, S.*

*3."Machine Learning for Cybersecurity: A Review" by Song, L., Wu, F., Chen, P., & Yang, Y.*

*which help us understand the practical ramifications of identifying malicious activities in Windows environments.*

***1.3 Project Context***

*Understanding the external systems, third-party interfaces, and collaborations required for the project's success is crucial. Interaction with virtual machine environments, the Atomic Red Team tool and sysmon logs. Key dependencies on individuals' time and access to resources further highlight the interconnected nature of the project.*

***External Systems:***

*Virtual Machine Environment for Attack Simulation*

*Windows Systems for Data Collection*

***Third-Party Interfaces/APIs:***

*Atomic Red Team for Attack Simulation*

*Windows Sysmon Logs for Real-Time Monitoring*

*Machine Learning Libraries and Tools for Model Development*

*Collaboration with Atomic Red Team for Attack Simulation*

*Key Dependencies:*

*Availability of Virtual Machine Environment*

*Access to Windows logs for Data Collection*

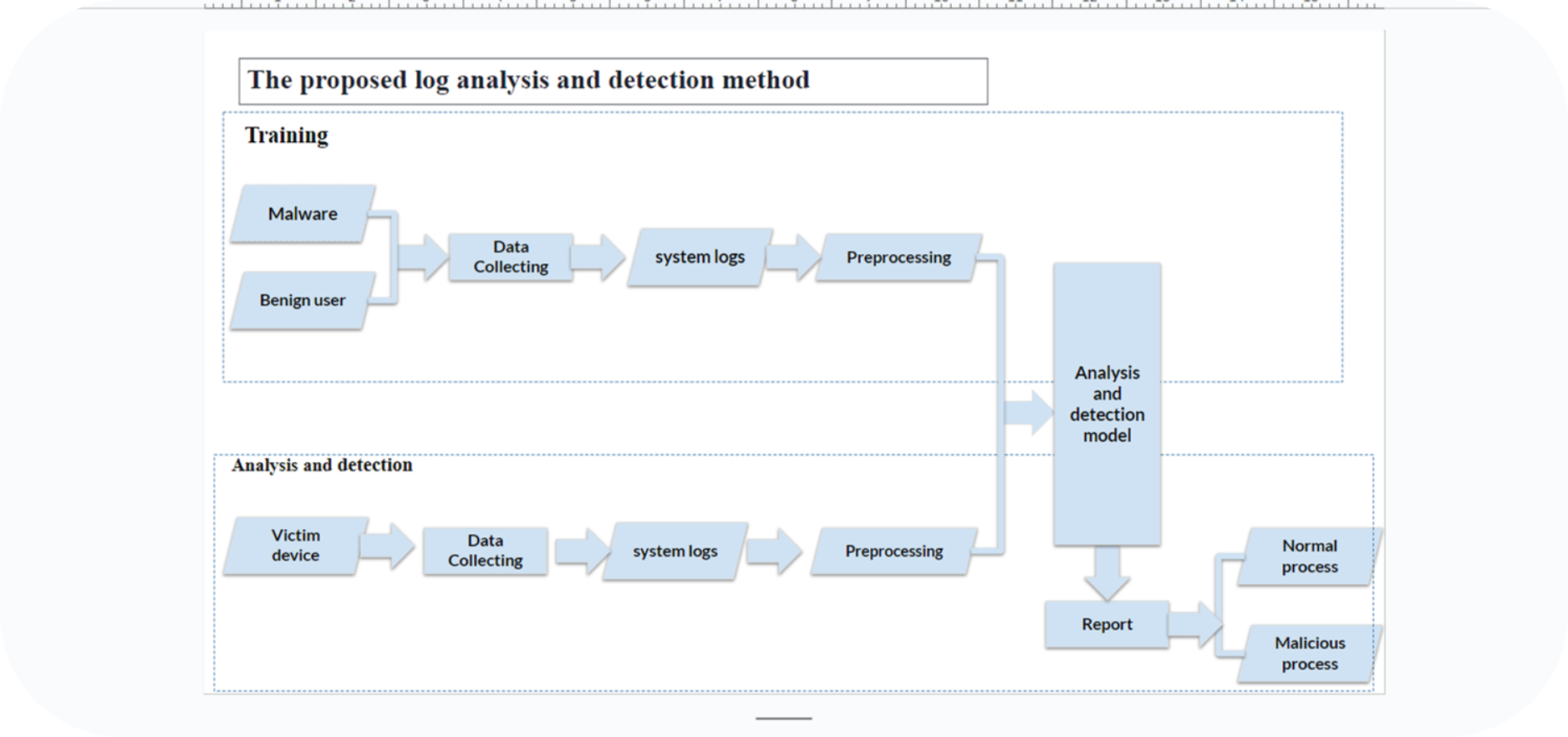
*The success of the project relies on the collaboration and support of external entities, emphasizing the interconnected nature of the cybersecurity landscape.*

***2. Design Overview***

*The system is designed to detect malicious behavior based on the analysis of audit logs. The conceptual design encompasses three main components: Data Collection, Model Development, and Deployment.*

*Architectural View:*

*The architectural view illustrates the flow of activities within the system. It begins with the simulation of attacks on a virtual machine, followed by the collection of logs. These logs undergo thorough preprocessing to extract critical features. The machine learning model is then trained using the preprocessed data. The trained model is saved and deployed to analyze new logs, providing insights into potential malicious activities.*

**

*Figure 1: The proposed design of the project*

***2.1 Requirements***

*Stakeholder Expectations:*

*Stakeholders expect a system that provides:*

*High Accuracy: The ability to accurately detect and classify malicious behavior.*

*Usability: A user-friendly interface for security analysts to interact with the tool.*

*Real-time Analysis: The deployment module should offer insights into potential threats in real-time.*

*Compatibility: The system should be adaptable to various log formats for seamless deployment.*

*End-User Needs:*

*End-users require:*

*User-Friendly Interface: An intuitive command-line interface for easy log analysis.*

*Comprehensive Log Analysis: The system should analyze logs from diverse sources, capturing various system events.*

*Efficient Model Deployment: A streamlined process for deploying the trained model for real-time log analysis.*

*Clear Feedback: The system should provide clear feedback on classification results to aid decision-making.*

***2.2 Detailed Design***

*To ensure the success of the "Malicious Behavior Detection Using Audit Logs" project, it is imperative to outline the specific requirements that align with the expectations of the stakeholders. These requirements are designed to satisfy end-user needs and address the core engineering problem of enhancing malicious behavior detection in Windows environments. The identified requirements encompass various aspects of the project, including functionality, performance, security, and usability.*

*1.Attack Simulator:*

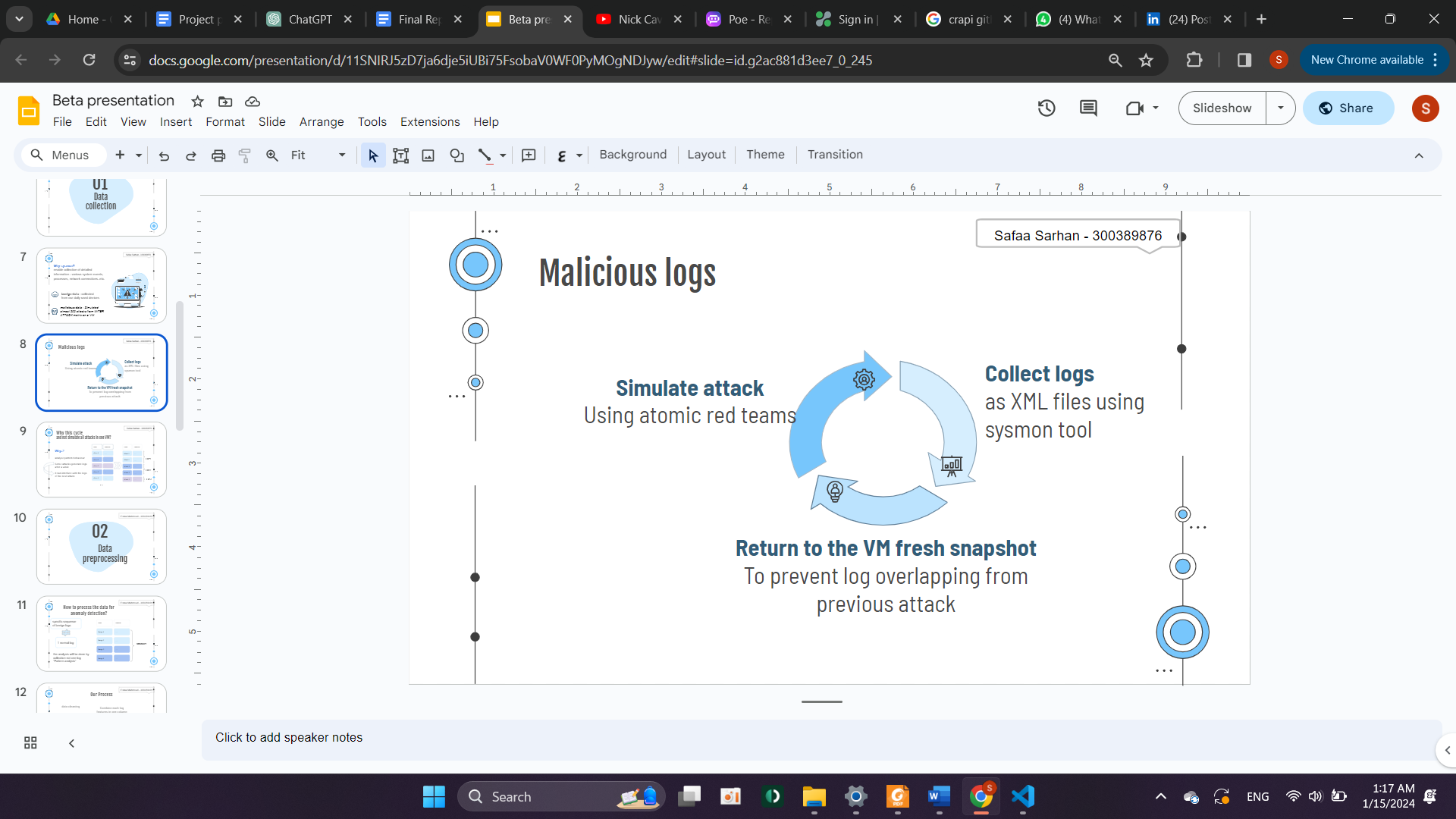
*Utilizes the Atomic Red Team framework for simulating diverse cyber attacks.*

*1. Log Collection module :*

*Collecting logs from Sysmon (System Monitor) is a crucial aspect of cybersecurity, providing valuable insights into the activities occurring on a Windows system. Sysmon is a powerful Windows Sysinternals tool developed by Microsoft, designed to monitor and log system activity, including process creation, network connections, file changes, and more. Analyzing these logs can help security professionals detect and respond to both benign and malicious activities on a system.*

*1.1.Collecting Benign Logs: from Daily Use Logs. process creation, network connection, file creation and modification, and registry modification.*

*1.2. Collecting Malicious Logs: Atomic Red Team with MITRE ATT&CK Framework To simulate and detect malicious activities, organizations often use tools like Atomic Red Team, which executes known adversarial techniques based on the MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework. Sysmon can capture logs related to these simulated attacks, allowing security teams to identify and respond to potential threats.*

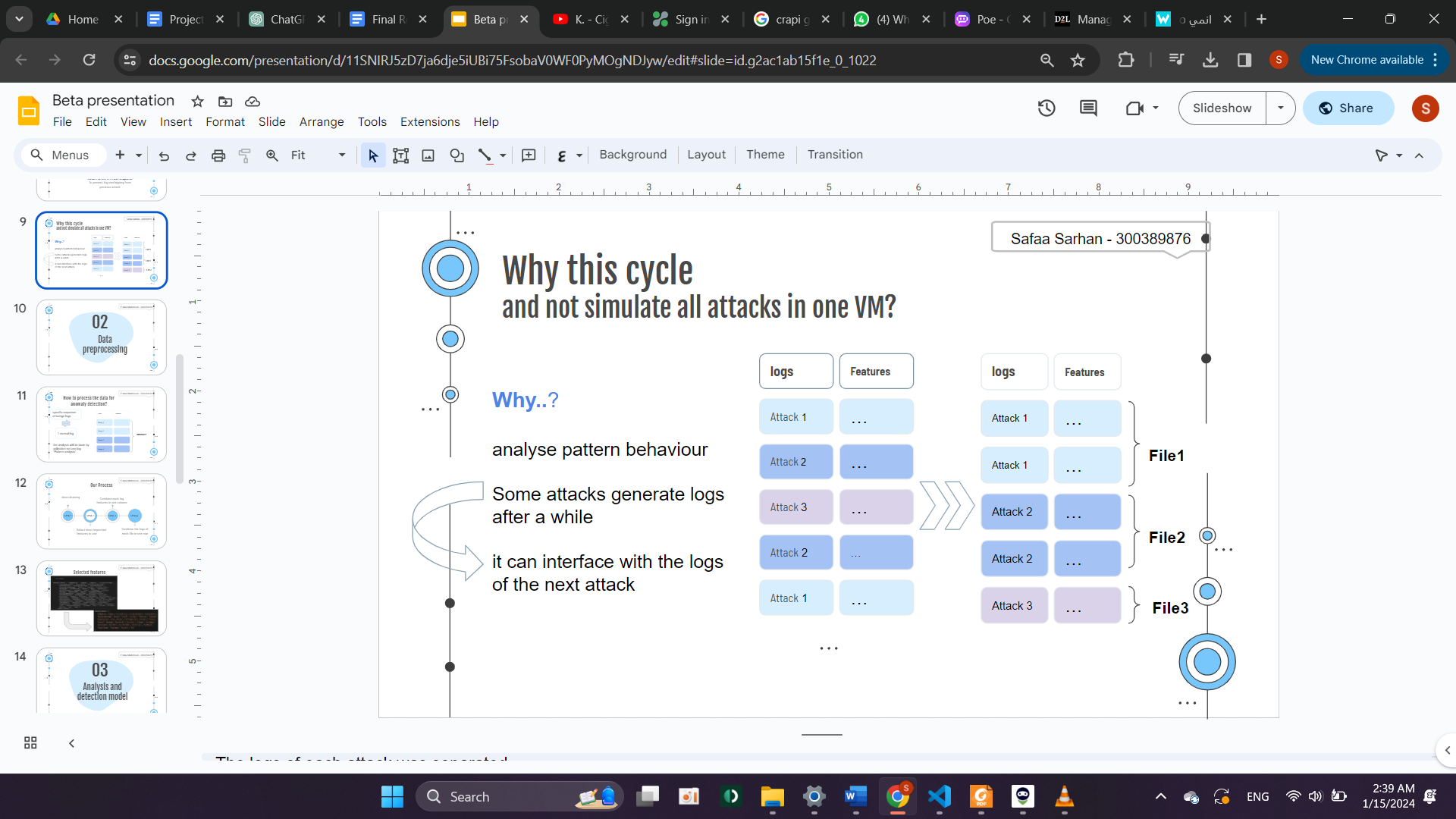
**

*Figure 2: The cycle of capturing the Sysmon logs*

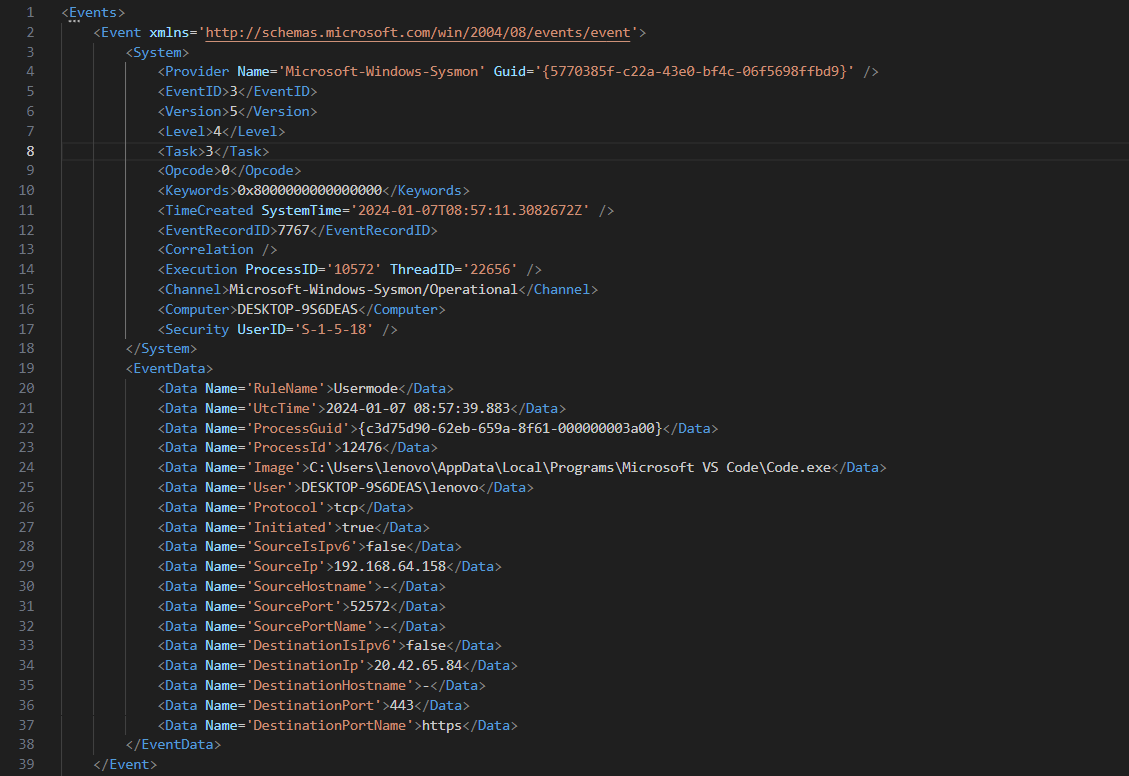
*The logs of each attack was separated*

*and not simulate all attacks in one VM to analyse the pattern and identify its behaviour*

*As Some attacks keeps generating logs after a while so it can interface with the logs of the next attack and pollute it*

**

*Figure 3: Explaining the collected logs how to be stored.*

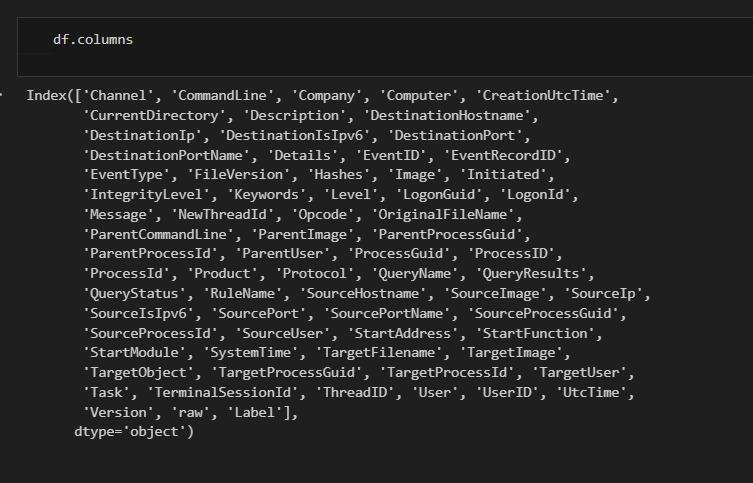
**

*Figure 4: Show the captured logs in XML format.*

*2. Preprocessing and Feature Extraction module:*

*When we save the extracted Sysmon logs, we save them as xml files. Then we convert them into CSV format using the pandas library.*

* *Selected features: In this preprocessing step, a thoughtful selection of features is made to distill the essential information from the Sysmon logs and facilitate the classification of behavior as either benign or malicious. The chosen features encompass a variety of aspects, ranging from process details to network-related information. By selecting these features, the preprocessing step aims to distill relevant information from the Sysmon logs, creating a more focused and informative representation that is conducive to subsequent analysis and classification of behaviors as either benign or malicious. This thoughtful feature selection enhances the efficiency of the subsequent machine learning or anomaly detection processes, streamlining the identification of potential security threats in the collected data.*

**

*Figure 5: The union of features from all logs files*

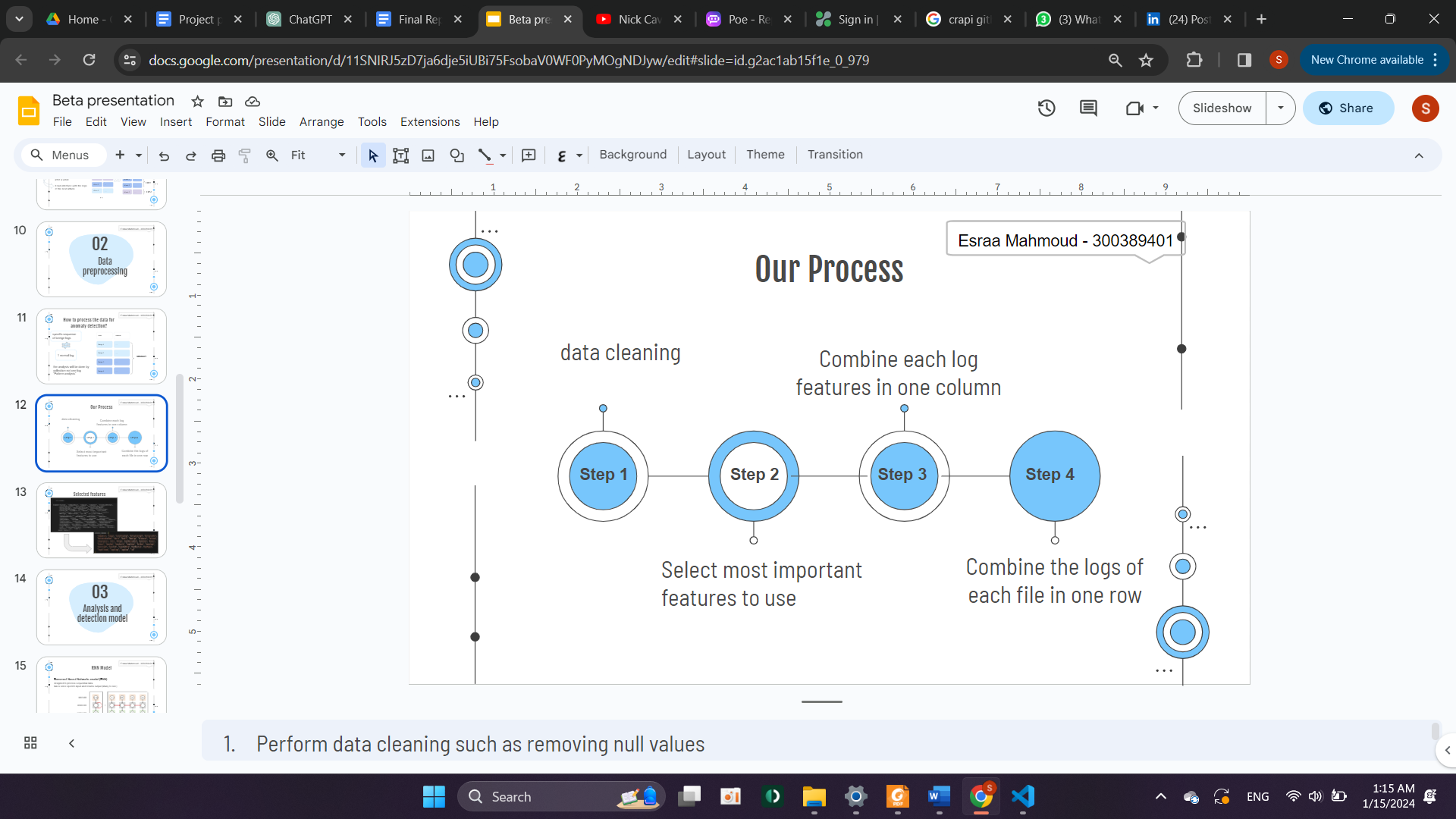
**

*Figure 6: The selected features that used in ML model*

*- Performing data cleaning such as removing null values: In the data preprocessing pipeline, handling null values is a critical step to ensure the integrity and reliability of the dataset. For the selected features in the Sysmon logs, such as "CommandLine," "Company," "CurrentDirectory," and others, it's essential to address any missing values appropriately. One common approach is to replace null values with zeros, especially when dealing with numerical features.*

* *Columns fixed length: applying specific length to all columns and padding them with zero.Then we combine all columns to make every Sysmon log file have one single column that has all selected features values.*
* *One feature vector: Then we performed one feature vector to be fed to the model which contains the values of all selected features of a log . Each element in the vector feature column represents the value of a specific feature for that instance.*
* *Thus we each log features values in one cell and then transpose the logs of each file in one row, So this row can represent the behaviour of an attack*

*The last step is combining all csv files that contain one single row that represents a collection of logs into one csv file to train the model.*

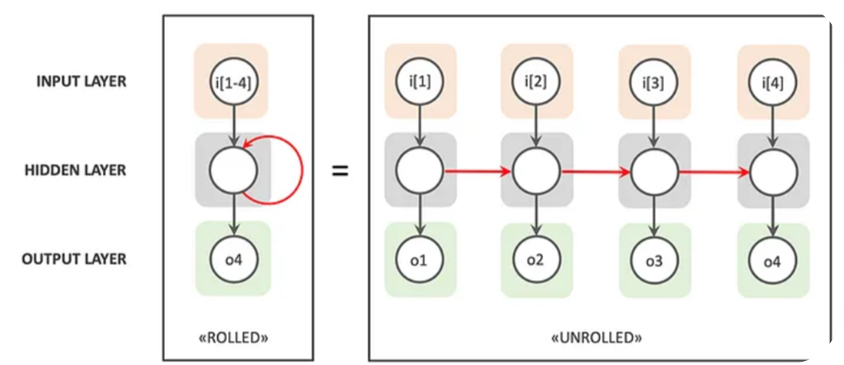
*- *

*Figure 7: The roadmap of logs data preprocessing*

*3. RNN Model Development:*

*Develop and implement a recurrent neural network (RNN) model for pattern analysis in Sysmon logs.*

*The code is for building a neural network model using the Keras library, which is a high-level neural networks API running on top of TensorFlow.*

**

*Figure 8: RNN Model explaination*

*Deployment Module:*

*Implements a command-line interface for user interaction.*

*Develops a unified script for log processing and model deployment.*

*Interaction and Interrelation:*

*The Attack Simulator feeds logs to the Data Collection Module.*

*Data Preprocessing extracts relevant features for the Feature Selection and Model Development phase.*

*The trained model from the Feature Selection and Model Development phase is used in the Deployment Module for real-time log analysis.*

***2.3 Implementation***

*The implementation phase of the project reached its final status with successful achievement of the defined goals. The primary objectives included developing an advanced system for detecting malicious behavior using audit logs, ensuring high accuracy in threat detection, and providing a CLI interface for log analysis.*

*The implementation process followed a systematic and structured approach to ensure the effective realization of the designed system. Key steps in the implementation included:*

*1. Attack Simulator Integration:*

*The integration with the Atomic Red Team framework for 300 attack simulation was successfully accomplished. The logs generated during these simulations served as a crucial dataset for training the machine learning model.*

*2. Data Collection:*

*The project employed efficient data collection strategies to gather a diverse set of logs for both malicious and benign activities. Approximately 300 attack simulations were conducted using the MITRE ATT&CK framework, and logs from everyday device usage were collected. This approach ensured a comprehensive dataset for model training.*

*3. Data Preprocessing:*

*A robust data preprocessing module was implemented using Pandas and NumPy libraries in Python. This module cleaned and structured raw log data, handling missing values, standardizing formats, and preparing the data for feature extraction. The preprocessing step played a crucial role in ensuring the quality of input data for the machine learning model.*

*4. Feature Selection and Model Development:*

*The selection of critical features for model training was performed meticulously. The Recurrent Neural Network (RNN) model, implemented using TensorFlow and Keras, demonstrated high efficiency in handling sequential data. The model architecture, consisting of an embedded layer, bidirectional LSTM layer, and dense layers with sigmoid activation, proved effective in capturing the nuances of user behavior.*

*The first layer is an Embedding layer. This layer is typically used for processing and embedding textual data. It takes three main parameters:*

*input\_dim: The size of the vocabulary, which is determined by the length of the word index created by the tokenizer.*

*output\_dim: The size of the vector space in which words will be embedded. In this case, it's set to 50.*

*input\_length: The length of input sequences, which is set to maxlen.*

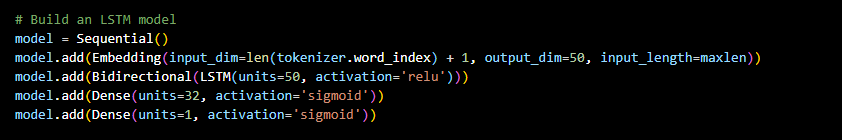
*The second layer is a Bidirectional LSTM (Long Short-Term Memory) layer. LSTM is a type of recurrent neural network (RNN) that is effective for processing sequences of data. Bidirectional indicates that the LSTM processes the input sequence in both forward and backward directions. It has one parameter:*

*units: The dimensionality of the output space, set to 50 in this case.*

*activation: The activation function used in the LSTM cells, set to 'relu'.*

*The third layer is a fully connected Dense layer with 32 units and a sigmoid activation function. This layer is often used for learning non-linear transformations from the input data.*

*The final layer is another Dense layer with a single unit and a sigmoid activation function. This layer is common in binary classification problems, where the model outputs a probability score between 0 and 1.*

**

*Figure 9: Model layers*

*5. Deployment:*

*The implementation included the development of a command-line interface for log analysis and model deployment. A unified script streamlined the process, allowing users to upload log files for analysis. The deployment module efficiently processed logs and provided clear classification results, indicating whether the behavior was malicious or benign.*

***Efficient Use of Tools and Technologies***

*The implementation phase made efficient use of various tools and technologies, contributing to the success of the project:*

*1. Python:*

*Python served as the primary programming language for the project. Its versatility and extensive libraries, including Pandas, NumPy, and TensorFlow, facilitated the development of scripts for attack simulation, data collection, preprocessing, and model training.*

*2. Atomic Red Team Framework:*

*The integration with the Atomic Red Team framework provided a standardized and controlled environment for simulating cyber attacks. The framework's API was leveraged to initiate and monitor attack simulations, ensuring the generation of realistic logs.*

*3. Machine Learning Libraries:*

*The project benefitted from the use of machine learning libraries, with TensorFlow and Keras playing a central role in developing and training the Recurrent Neural Network model. These libraries streamlined the implementation of complex neural network architectures.*

*Conclusion of the Implementation Phase*

*In conclusion, the implementation phase of the project successfully translated the design into a functional system. The systematic approach, efficient use of tools and technologies, and the integration of diverse modules contributed to the project's overall success. The implemented system demonstrates the capability to analyze logs in real-time, detect malicious behavior accurately, and provide valuable insights for threat detection and incident response.*

***2.4 Testing***

***2.4.1 Data Plan***

*Data Collection:Simulated almost 300 attacks using the MITRE ATT&CK framework.*

*Collected logs from diverse sources, including both simulated attacks and benign data.*

***2.4.2 Validation & Verification***

*Verification: Ensured the system met design specifications through code reviews and testing.*

*Checked the system against predefined criteria for accuracy, usability, and real-time analysis.*

*Validation: Utilized simulations to predict potential faults or gaps in the system.*

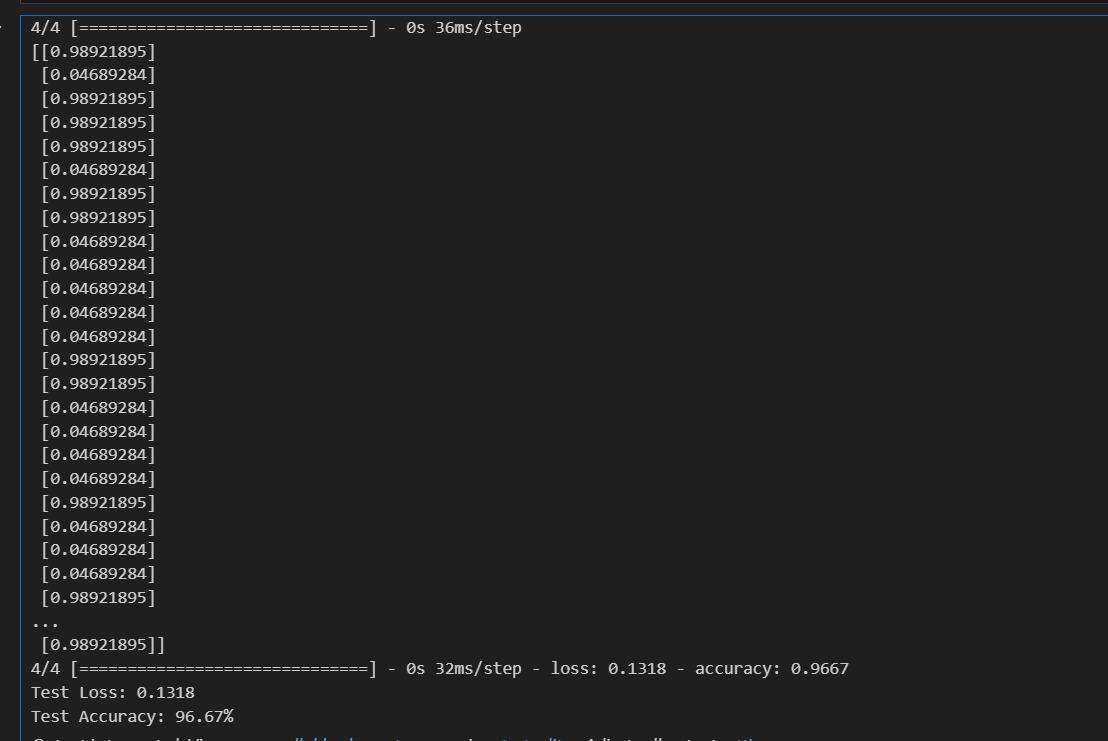
*Ran test cases to ensure the system met operational needs, addressing both initial design requirements and ongoing specifications.*

*The iterative testing process involved continuous validation and verification, ensuring the system's resilience and effectiveness in detecting malicious behavior.*

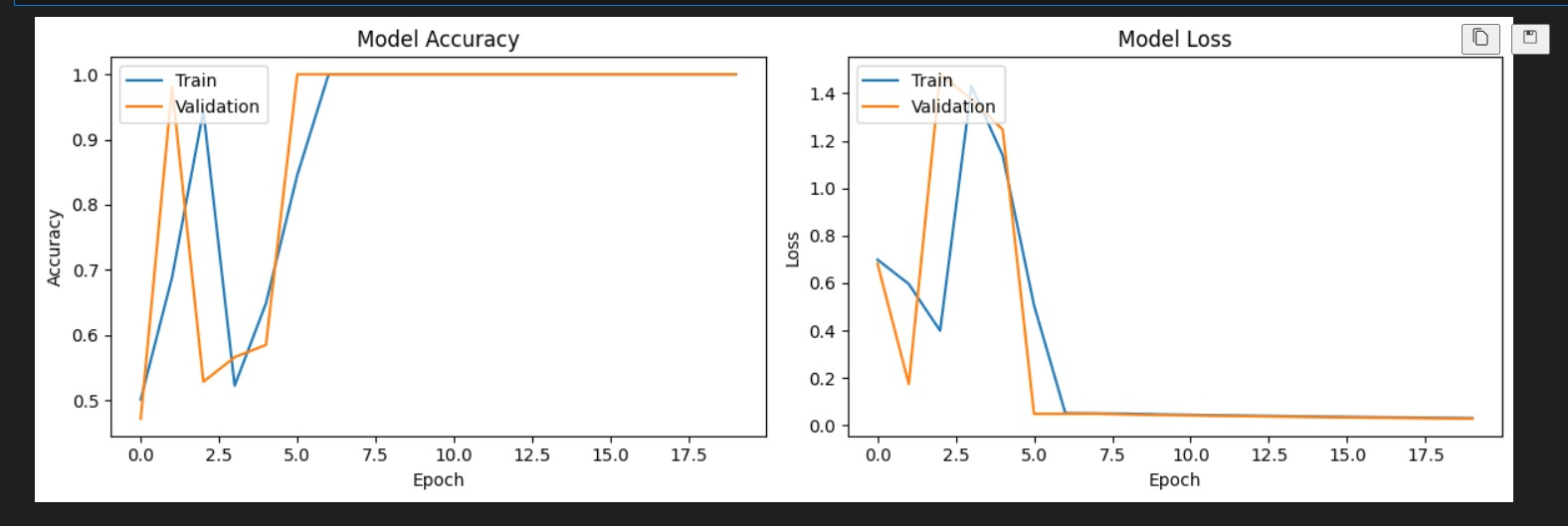
***3. Overall Results and Analysis***

*3.1 Project Evaluation*

*The project achieved success in developing a system with high accuracy in detecting malicious behavior in audit logs. The real-time analysis capability, facilitated by the deployment module, provided security analysts with timely insights into potential threats.*

**

*Figure 10: Test accuracy*

**

*Figure 11: validation accuracy*

*The successful integration of machine learning, particularly the Recurrent Neural Network model, with the simulated attacks using the Atomic Red Team framework, demonstrated the effectiveness of combining diverse technologies for enhanced threat detection.*

*3.1.2 Challenges and Lessons Learned*

*Data Preprocessing Complexity:*

*The complexity of preprocessing diverse logs from simulated attacks posed challenges. Some attacks generated logs in varied formats, necessitating additional efforts to standardize and extract meaningful features.*

*Dependency on External Systems:*

*The project's success was contingent on access to virtual machine environments for attack simulation. Coordinating with external entities for collaboration and resource access added an extra layer of complexity.*

*Iterative Model Refinement:*

*Iterative refinement of the machine learning model was crucial. While the RNN architecture showed promising results, initial versions required fine-tuning to achieve optimal accuracy and minimize false positives.*

***4. Deployment Plan***

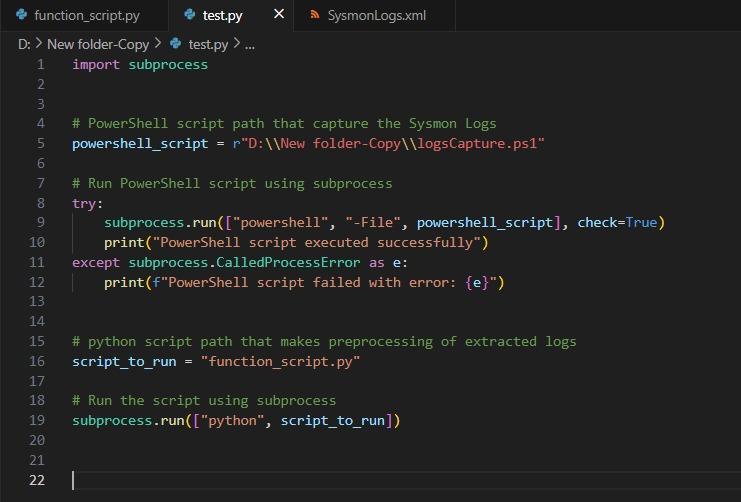
*In deploying our anomaly detection model, we've devised a comprehensive strategy accommodating both user-initiated searches and real-time monitoring of Windows Sysmon logs. Users can seamlessly search for anomalies by either uploading single log files or leveraging Sysmon logs from their environment, facilitated by a unified script adept at handling various log formats, including those from Windows systems (Sysmon Logs) and custom sources. This script outputs a standardized CSV file, serving as the input for the anomaly detection model during the detection phase. The trained model, stored in a pickle file, efficiently processes new logs, distinguishing between malicious and benign activities. Our command-line interface ensures a straightforward experience for both end-users and security analysts, enabling them to test logs with minimal installation effort*

#### *4.1 Utilizing the Anomaly Detection Model:*

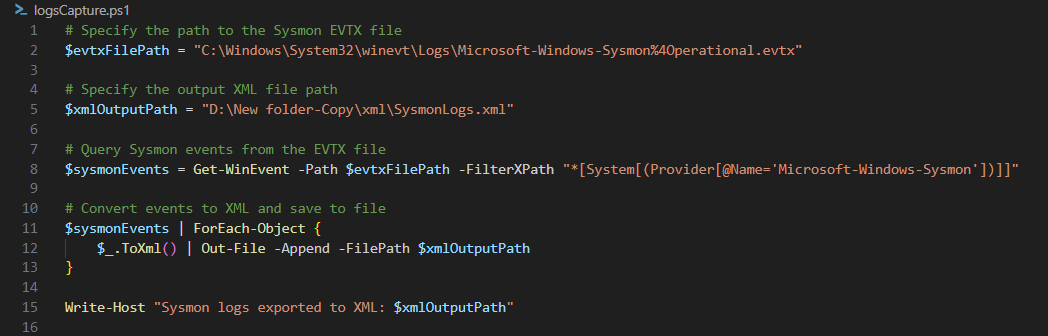
*After completing the preparation of our anomaly detection model, our focus shifts to an effective deployment strategy catering to both user-initiated searches and continuous monitoring of real-time Windows Sysmon logs.*

#### *User-Initiated Anomaly Searches:*

* *User Options:*
  + *Users can search for anomalies by either uploading a single log file or utilizing Windows Sysmon logs from their environment.*
* *Streamlining Deployment:*
  + *To simplify the process, we have implemented a unified script capable of handling various log formats. This includes logs exported from Windows systems or captured from custom log sources.*
* *Output Format:*
  + *The script outputs a standardized CSV file, serving as the input for the anomaly detection model. This CSV file represents the final preprocessed data shape.*

**

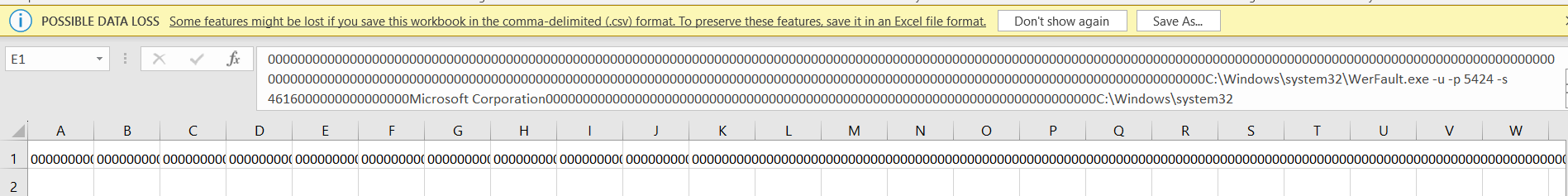
*Figure 12: ps script to capture real-time logs*

**

*Figure 13: capturing real time logs and save them.*

#### *4.2 Detection Phase:*

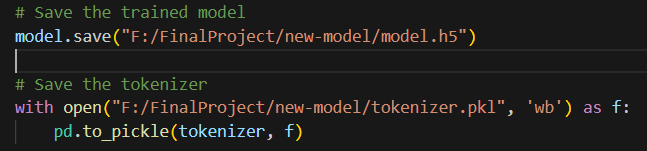
* *XML Log Processing:*
  + *XML logs typically require significant effort for processing.*
* *Deployment Streamlining:*
  + *Our deployment strategy involves streamlining this process with a unified script capable of handling various log formats.*
* *Model Input:*
  + *The script outputs a standardized CSV file, which serves as the input for the anomaly detection model during the detection phase.*

**

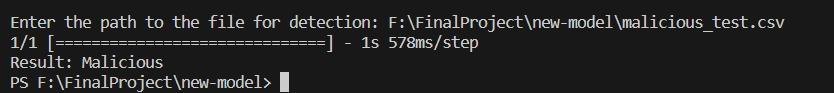
*Figure 14: data after preprocessing to be tested*

#### *4.3 Model Deployment:*

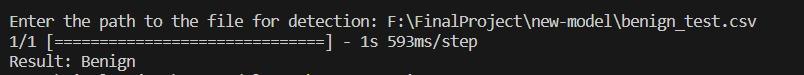
* *Efficient Storage:*
  + *The trained model is saved in a pickle file for efficient deployment.*
* *Prediction Process:*
  + *The saved model is fed with new logs during the detection phase, distinguishing between malicious and benign activities.*

**

*Figure 15: Saving the trained model*

**

*Figure 16: Testing malicious data*

**

*Figure 17: Testing benign data*

#### *4.4 User and Analyst Experience:*

* *Command Line Interface:*
  + *Our command-line interface provides a seamless experience for end-users and security analysts.*
  + *Users can test logs from their environments with minimal installation effort using a single script.*

### *4.5 Ensuring Operational Readiness:*

* *Pre-Deployment Checklist:*
  + *Conduct thorough testing to ensure the system/application is ready for distribution.*
  + *Verify that all dependencies, libraries, and configurations are correctly set up in the operational environment.*
* *Availability Requirements:*
  + *Ensure that necessary resources, such as computational power and memory, are available in the operational environment to support the application's requirements.*
* *Maintainability Considerations:*
  + *Implement proper logging mechanisms to facilitate easy troubleshooting.*
  + *Provide clear documentation for end-users and administrators regarding installation, configuration, and troubleshooting processes.*
* *User-Friendly Interface:*
  + *Design the user interface with simplicity in mind to create a smooth experience.*
  + *Offer user-friendly error messages and instructions for better user understanding.*
* *Regular Updates and Support:*
  + *Establish a system for regular updates to address potential security vulnerabilities and improve functionality.*
  + *Provide a support mechanism for end-users to report issues and receive timely assistance.*
* *Monitoring and Alerts:*
  + *Implement monitoring mechanisms to track system/application performance.*
  + *Set up alerts for any anomalies or errors to ensure prompt attention to potential issues.*
* *Scalability Planning:*
  + *Consider scalability requirements and design the system/application architecture to handle potential increases in data volume and user activity.*

***5. Conclusions and Future Works***

*The project successfully developed a comprehensive solution for the detection of malicious behavior using Sysmon logs. Key accomplishments include the integration of machine learning with simulated attacks, achieving high accuracy in threat detection, and providing a user-friendly interface for security analysts.*

*In conclusion, this project has laid a foundation for advancing the field of cybersecurity by effectively combining simulated attacks, machine learning, and real-time log analysis. The developed system not only meets the initial goals but also provides valuable insights into the intersection of cyber threat simulation and anomaly detection.*

*5.1 Future Works and Research Paths*

*Future plans involve generating insightful reports and enhancing the model's capabilities by incorporating more threat types. Ensuring operational readiness entails testing, availability verification, maintainability considerations, a user-friendly interface, regular updates, monitoring mechanisms, and scalability planning to guarantee a smooth end-user experience and a robust operational environment*

*Insightful Reports:*

*Future enhancements include generating insightful reports for end-users and security operation centers.*

*These reports aid in the detection of suspicious behavior within operational environments.*

*Model Enhancement:*

*Ongoing efforts involve enhancing the model's capabilities by incorporating more threat types.*

*This ensures a broader identification range for diverse log entries.*

*5.2. Integration with SIEM Systems:*

*Explore the integration of the developed system with Security Information and Event Management (SIEM) systems for enhanced cybersecurity operations.*

*5.3. Collaboration and Industry Adoption:*

*Seek collaboration with cybersecurity professionals and organizations for real-world testing and industry adoption.*

*5.4. Unexplored Questions and Research Opportunities*

*5.4.1. Dynamic Threat Modeling:*

*Investigate dynamic threat modeling to adapt the system's model to changing attack patterns and behaviors.*

*5.4.2. Explainable AI in Cybersecurity:*

*Explore the incorporation of explainable AI techniques to enhance the interpretability of the machine learning model for security analysts.*

*5.4.3. Scalability and Performance Optimization:*

*Research and implement techniques for optimizing the system's performance and scalability to handle large volumes of logs efficiently.*

*5.5. Improving Solution and Additional Requirements*

*5.5.1. User Feedback Mechanism:*

*Incorporate a user feedback mechanism to enhance the system's learning and adaptability based on real-world usage patterns.*

*5.5.2. Multi-Modal Log Analysis:*

*Consider expanding the system's capabilities to analyze logs from multiple sources and modalities for a more comprehensive threat detection.*

*5.5.3. Threat Intelligence Integration:*

*Integrate threat intelligence feeds to enhance the system's ability to recognize and respond to emerging threats proactively.*

*5.6 Motivation for Research Path*

*The research path taken in this project is motivated by the imperative need for advanced cybersecurity solutions that can effectively adapt to the dynamic nature of cyber threats. By combining simulation, machine learning, and real-time analysis, the developed system addresses the complexities of modern cyber landscapes. The motivation lies in the continuous pursuit of innovation to stay ahead of cyber adversaries and provide robust, scalable, and efficient solutions for cybersecurity challenges.*

*In conclusion, the journey of this project not only yields tangible results in the form of a functional system but also opens the door to a broader research landscape where the intersection of cyber threat simulation, machine learning, and real-time analysis can lead to transformative advancements in cybersecurity.*

***6. References***

***[1] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda. Panorama: capturing system-wide information flow for malware detection and analysis. In Proceedings of the 14th ACM Conference on Computer and Communications Security, pages 116–127. ACM, 2007***

***[2] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda. Beehive:Large-scale log analysis for detecting suspicious activity in enterprise networks. In Proceedings of the 29th Annual Computer Security Applications Conference, pages 199–208. ACM, 2013.***

***[3] A. Moser, C. Kruegel, and E. Kirda. Limits of static analysis for malware detection. In Proceedings of the 23rd Computer Security Applications Conference, pages 421–430, 2007.***

***[4] E. Gandotra, D. Bansal, and S. Sofat. Malware analysis and classification: A survey. Journal of Information Security, 5(02):56, 2014.***

***[5] Nikam, U.V.; Deshmuh, V.M. Performance evaluation of machine learning classifiers in malware detection. In Proceedings of the 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 23–24 April 2022; pp. 1–5. [CrossRef]***

***[6] Sethi, K.; Kumar, R.; Sethi, L.; Bera, P.; Patra, P.K. A novel machine learning based malware detection and classification framework. In Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 3–4 June 2019; pp. 1–13.***

***[7] Sharma, S.; Krishna, C.R.; Sahay, S.K. Detection of advanced malware by machine learning techniques. In Proceedings of the SoCTA 2017, Jhansi, India, 22–24 December 2017.***

***[8] Gibert, D.; Mateu, C.; Planes, J.; Vicens, R. Using convolutional neural networks for classification of malware represented as images. J. Comput. Virol. Hacking Tech. 2019, 15, 15–28. [CrossRef]***

***[9] Dahl, G.E.; Stokes, J.W.; Deng, L.; Yu, D.; Research, M. Large-scale Malware Classification Using Random Projections And Neural Networks. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing-1988, Vancouver, BC, Canada, 26–31 May 2013; pp. 3422–3426***

***[10] Agarkar, S.; Ghosh, S. Malware detection & classification using machine learning. In Proceedings of the 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Gunupur Odisha, India, 16–17 December 2020; pp. 1–6.***

***[11]*** [***https://github.com/redcanaryco/atomic-red-team***](https://github.com/redcanaryco/atomic-red-team)

***[12]*** [***https://github.com/redcanaryco/invoke-atomicredteam***](https://github.com/redcanaryco/invoke-atomicredteam)