Proposal for Minor Project of Bachelor of Computer Engineering

Network Intrusion Detection System

Submitted by:

Prajjwal Adhikari [BEC-2020-20]

Rozal Dahal [BEC-2020-27]

Aman Poudel [BEC-2020-03]

Raj Gurung [BEC-2020-23]

Submitted To:

United Technical College

Faculty of Science and Technology

Pokhara University, Nepal

Jan, 2024

Abstract

The escalating growth of the Internet and communication networks has led to an exponential increase in transmitted data, making networks susceptible to evolving cyber threats. In response, this project proposes the development of a sophisticated Network Intrusion Detection System (NIDS) that harnesses the power of advanced machine learning algorithms. Emphasizing real-time monitoring and anomaly-based detection, the NIDS aims to bolster network security by addressing the limitations inherent in existing intrusion detection mechanisms. By integrating Software Defined Networking (SDN) and deep learning technologies, the system aspires to provide efficient response and reporting mechanisms tailored to small-scale networks. The project's methodology involves meticulous data collection through strategically deployed sensors and supplementary datasets. The processed data undergoes feature extraction, and the Random Forest classifier is employed for training the intrusion detection model. Development tools encompass Python, C/C++, Java, and various libraries and frameworks supporting packet capture, deep packet inspection, machine learning, database management, and more. The expected outcome includes a robust NIDS capable of real-time intrusion detection, efficient anomaly-based identification, and prompt response mechanisms. The impact of the project lies in fortifying network security, particularly in Small Office Home Office (SOHO) environments and LAN networks. By seamlessly integrating signature and anomaly detection, the proposed NIDS contributes to a unified and intelligent defense against a diverse range of cyber threats.

Keywords: Network Intrusion Detection System, Machine Learning, Cybersecurity, Anomaly Detection, Real-time Monitoring, SDN, Deep Learning.

Table of Content

List of Figures

Abbreviation and Acronyms

CNN	Convolution Neural Network
CPU	Central Processing Unit
DARPA	Defense Advance Research Project Agency
DL	Deep Learning
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Protection System
KNN	K-Nearest Neighbor
LAN	Local Area Network
ML	Machine Learning
NIDS	Network Intrusion Detection System
RAM	Random Access Memory
SDN	Software Defined Networking Technology
SOHO	Small Office Home Office
SQL	Structured Query Language
SSL	Secure Socket Layer
TCP	Transfer Control Protocol
TLS	Transport Layer Security

Chapter 1: Introduction

1.1 Background

In the dynamically evolving realm of global connectivity and digital dependence, ensuring the security of computer networks has become an imperative undertaking. The exponential growth of the Internet and communication networks has given rise to a sophisticated breed of cyber threats, where attackers employ innovative methods to compromise and manipulate sensitive information [1]. Organizations are confronted with an ever-expanding threat landscape, ranging from traditional intrusion attempts to intricate zero-day exploits, challenging the resilience of their networks.

A cornerstone in the defense against these cyber threats is the Network Intrusion Detection System (NIDS), a specialized security solution designed to vigilantly monitor and analyze network traffic. However, the inadequacies of existing intrusion detection mechanisms are increasingly apparent. Signature-based detection, a fundamental approach, falls short in identifying novel or zero-day attacks, while anomaly-based methods grapple with elevated false-positive rates [3].

The scalability of intrusion detection systems further becomes a concern as networks grow in complexity. The absence of a unified and intelligent NIDS that seamlessly integrates signature and anomaly detection, adapts to emerging threats, and ensures scalability impedes organizations' proactive defense against a diverse range of cyber threats [1].

Motivated by the urgent need for advanced network security measures, this project proposes the development of a sophisticated NIDS. Harnessing the capabilities of machine learning algorithms, notably emphasizing the Random Forest classifier [2], the project aims to overcome the limitations of traditional intrusion detection mechanisms. The incorporation of Software Defined Networking (SDN) technology and deep learning techniques enhances the NIDS's ability to respond efficiently to emerging threats in real-time.

The envisaged NIDS is tailored for implementation in Small Office Home Office (SOHO) devices and small-scale networks, such as Local Area Networks (LANs). By providing a complementary second line of defense alongside existing preventive measures like firewalls, the NIDS seeks to actively monitor traffic, identify anomalies, and bolster overall cybersecurity.

In summary, this project endeavors to contribute to fortifying network security against an ever-evolving cyber threat landscape, safeguarding the integrity and confidentiality of crucial information traversing modern communication networks [1] [2].

1.2 Statement of Problem

In contemporary network security, the surge in sophisticated cyber threat poses a substantial challenge to the resilience of organizational networks. Existing intrusion detection mechanism, while fundamental, encounter limitations in effectively addressing the dynamic nature of modern cyber threats. Signature-based detection often fall short in identifying novel or zero-day attacks, and anomaly-based approaches may struggle with high false-positive rates. [3]

Additionally, the scalability of intrusion detection systems become a concern as network expand in complexity. Furthermore, the lack of a unified and intelligent Network Intrusion Detection System (NIDS) that seamlessly integrates signature and anomaly detection, adapts to emerging threats and ensures scalability, hampers the ability of organization to proactively safeguard their networks against a diverse range of cyber threats [1].

1.3 Objective

Our project Network Intrusion Detection System is going to be develop to meet the following objectives:

- To have Real-time Monitoring of network packets in system
- To build the features of Anomaly-Based Detection of network intrusion
- To achieve the Response and Reporting after detection of intrusion at system

1.4 Application

A NIDS plays a pivotal role in bolstering network security by providing a second line of defense. While Firewalls and others preventive measures are crucial, they may not catch all

potential threats. NIDS complements these preventive measures by actively monitoring traffic and identifying anomalies or patterns indicative of malicious activity.

1.5 Limitations and Scope

This project is built with intention to be implemented in SO-HO devices or the small networks like LAN network to detect the packets lively if possible else to analyze the network packets to detect the intrusion in the system.

This project is limited to monitor the network traffic working as the second line of defense to coordinate with firewalls and monitor the overall traffic either in real-time or nearly real time in network to identify the anomalies.

Chapter 2: Literature Review

1.1 Introduction

With intention to meet the objective we have set for the project accomplishment we have decide to go through the different paperwork that have been made in the past history. Network Intrusion Detection System is a system that will be able to do monitor network traffic and report the anomalies. A lot of research has been done for this project by our team on Network Intrusion Detection System and here are some of the study's results.

1.2 Case Study

1. **Network Intrusion Detection System Using Neural Network**

   This research introduces a neural network-driven approach for detecting internet-based attacks on computer networks. Intrusion Detection Systems (IDS) have been developed to anticipate and prevent both existing and upcoming cyber threats. The method relies on neural networks to recognize and forecast abnormal behaviors within the system, specifically employing feedforward neural networks with the backpropagation training algorithm. The study utilized training and testing data extracted from the Defense Advanced Research Projects Agency (DARPA) intrusion detection evaluation datasets. The empirical findings, based on actual data, demonstrated encouraging outcomes for the detection of intrusion systems through the application of neural networks [4].

2. **Survey on SDN based network intrusion detection system using machine learning approaches**

   Software Defined Networking Technology (SDN) presents an opportunity to efficiently identify and monitor network security issues by virtue of its programmable features. Recently, Machine Learning (ML) approaches have been integrated into SDN-based Network Intrusion Detection Systems (NIDS) to safeguard computer networks and address security challenges. A series of advanced machine learning techniques, particularly deep learning technology (DL), is emerging within the SDN framework. This survey explores recent research on ML

methods that leverage SDN for NIDS implementation, with a specific focus on the application of deep learning techniques. Additionally, the survey covers tools applicable for developing NIDS models within the SDN environment. It concludes with a discussion on ongoing challenges in implementing NIDS using ML/DL and outlines potential future research directions [5].

3. A high-performance network intrusion detection system

This paper presents a new approach for network intrusion detection system based on concise specification that characterize normal and abnormal network packet sequences. They have specification language geared for a robust network intrusion detection by enforcing a strict type discipline via a combination of static and dynamic type of checking. Unlike most of other approaches in network intrusion detection this approach can easily support new network protocols as information relating to the protocols are not hard-coded into the system instead they have added suitable type definitions in the specifications and define intrusion patterns on these types. [6]

4. A Study of NIDS using Artificial Intelligence/ Machine Learning

The surge in Internet and communication growth has led to a substantial increase in transmitted data, attracting continuous novel attacks from adversaries seeking to steal or corrupt this valuable information. Intrusion Detection Systems (IDS) play a crucial role in detecting such intrusions by examining network traffic. Despite numerous research efforts to enhance IDS solutions, there remains a need for improvement in detection accuracy and reduction of false alarm rates, especially against zero-day attacks. Machine learning algorithms have gained popularity for efficient and accurate network intrusion detection. This paper introduces the concept of IDS, presents a taxonomy of machine learning methods, discusses key metrics for IDS assessment, and reviews recent ML-based IDS solutions, highlighting their strengths and weaknesses. It also examines datasets used in these studies, discusses result accuracy, and concludes with observations, research challenges, and future trends in the field. [7]

With the study of different project, surveys and case studies a conclusion is made that using the machine learning technique with predefined sets of anomalies should be train to achieve the network intrusion detection system with use of algorithms like KNN, Random Forest Classifiers etc for our project of Network Intrusion Detection System.

Chapter 3: Methodology

3.1 Flow of project

A project flow diagram is a visual representation that outlines the sequence and dependencies of tasks or activities in a project. It illustrates the flow of work from start to finish, highlighting the order and relationships between different project components. The diagram helps stakeholders understand the project's timeline, milestones, and critical paths, facilitating effective project planning, coordination, and communication.

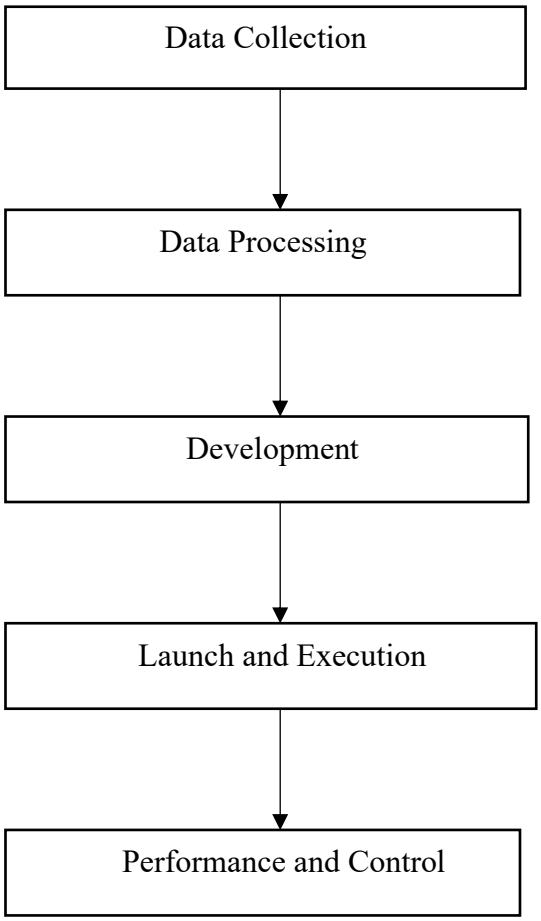

Figure 1 : Project Flow Diagram (Group Study)

3.1.1 Data Collection

The foundation of our Network Intrusion Detection System lies in the meticulous collection of network traffic data. To achieve this, we will strategically deploy sensors within the network infrastructure, tasked with the real-time capture of packet information. Utilizing widely recognized tools such as Wireshark, we will conduct live monitoring sessions to

capture primary source data, ensuring the authenticity and relevance of the information gathered. Additionally, we will supplement this primary data with pre-collected datasets obtained from reputable sources on the internet. These datasets, drawn from various network environments, will further diversify our dataset, providing a comprehensive representation of potential network intrusion scenarios. The collected data, consisting of attributes such as source and destination IP addresses, protocols utilized, and packet sizes, will serve as the raw material for our intrusion detection system. This comprehensive approach to data collection, incorporating both primary and supplementary sources, ensures that our system is equipped with a diverse and representative dataset, enhancing the efficacy of the subsequent machine learning model.

3.1.2 Data Processing

Following the comprehensive data collection phase, the next critical step is data processing, where the gathered information undergoes meticulous extraction of relevant features. Feature extraction involves identifying key characteristics, such as packet timestamps, payload content, and traffic patterns, that are instrumental for the machine learning model. The dataset, enriched with both primary source data from live monitoring sessions using Wireshark and pre-collected datasets from reputable internet sources, is then divided into training and testing sets. A significant portion is allocated for training the Random Forest classifier, chosen for its exceptional ability to process high-dimensional data and capture intricate relationships within the dataset.

The Random Forest algorithm excels in handling the dynamic and diverse nature of network traffic. During the training phase, the algorithm constructs multiple decision trees on different subsets of the training data, preventing overfitting and ensuring adaptability to evolving network patterns. Hyperparameter tuning further refines the model's performance, optimizing its ability to discern normal network behavior from potential intrusions. Subsequently, the trained Random Forest model undergoes rigorous validation using the testing dataset to confirm its proficiency in accurately identifying and classifying network intrusions.

This meticulous approach to data processing, leveraging both primary and supplementary sources, coupled with the power of the Random Forest classifier, forms the robust foundation of our Network Intrusion Detection System, ensuring accurate and reliable intrusion detection capabilities.

3.1.3 Development

Development tools for projects encompass a wide range of software applications, frameworks, and utilities that aid in the creation and management of project deliverables. These tools streamline development processes, enhance collaboration, and provide features for code editing, version control, testing, and project tracking.

1. **Programming Languages:**
   - Python: Widely used for its simplicity and extensive libraries, Python is suitable for implementing various components of a NIDS [8].
   - C/C++: Ideal for low-level operations and performance-critical tasks within the NIDS [9].
   - Java: Suitable for developing cross-platform applications and components [10].

2. **Packet Capture and Analysis:**
   - Scapy: A powerful Python library for capturing, manipulating, and sending network packets [11].
   - Wireshark: A widely used packet analysis tool that can be integrated into the development process for testing and validation [12].

3. **Deep Packet Inspection (DPI):**

- Snort: An open-source NIDS that includes a DPI engine for analyzing network traffic and detecting signatures of known attacks [13].
- Suricata: An open-source IDS/IPS engine that supports multi-threading and is designed for high-performance deep packet inspection [13].

4. **Machine Learning and Anomaly Detection:**

- Scikit-learn: A machine learning library for Python, useful for implementing supervised and unsupervised learning algorithms [14].
- TensorFlow and PyTorch: Deep learning frameworks that can be employed for building neural network-based anomaly detection models [14].

5. **Database Management:**
   - MySQL or PostgreSQL: Relational database management systems that can be used to store configuration data, logs, and other relevant information [15].

- Elasticsearch: A distributed search and analytics engine, useful for storing and querying large volumes of log data [16].

6. **Web Frameworks (Optional):**
   - Django or Flask: If a web-based user interface is part of the NIDS, these Python web frameworks can be used for rapid development [17].

7. **Data Visualization:**
   - Matplotlib and Seaborn: Python libraries for creating static, animated, and interactive visualizations of data [18].
   - Kibana: A data visualization tool often used with Elasticsearch for exploring and visualizing log data [18].

8. **Security Protocols and Encryption:**
   - SSL/TLS: For securing communication channels and encrypting data [19].
   - IPsec: A suite of protocols for securing Internet Protocol (IP) communications [19].

9. **Logging and Reporting:**
   - Syslog: A standard protocol for sending log messages in a network [20].
   - Logstash: Log data processing tool that can collect, parse, and enrich log data [20].

10. **Network Protocols and Standards:**
    - TCP/IP Stack: Understanding of the TCP/IP protocol suite is fundamental for analyzing network traffic [21].

11. **Version Control:**
    - Git: A distributed version control system for tracking changes in the source code [22].

3.1.4 Launch and Execution

After completion of development phase our product will be launched at United Technical College, Bharatpur-11, Chitwan.

3.1.5 Performance and Control

Project will be tested for its performance and will be maintained accordingly.

3.2 System Design

This portion of paper contains the architecture of Network Intrusion Detection System based on the information collected in earlier portion of document. It contains diagrams like System Diagrams, Flow Charts etc.

3.2.1 System Diagram

Picture Shown below is a general view of Network Intrusion System Implementation segment to monitor the traffic.

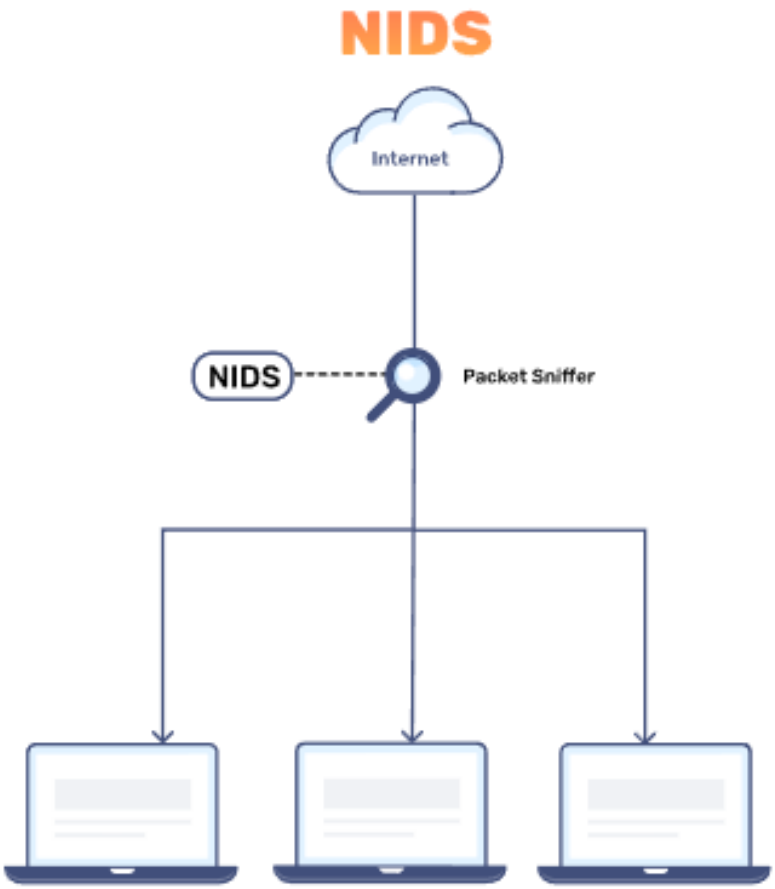

Figure 2: NIDS (Bunny.net)

3.2.2 ER diagram

An ER diagram, short for Entity-Relationship diagram, is a graphical representation of the entities, attributes, and relationships within a database system. It provides a visual tool for designing and understanding the structure and relationships of the data in a database.

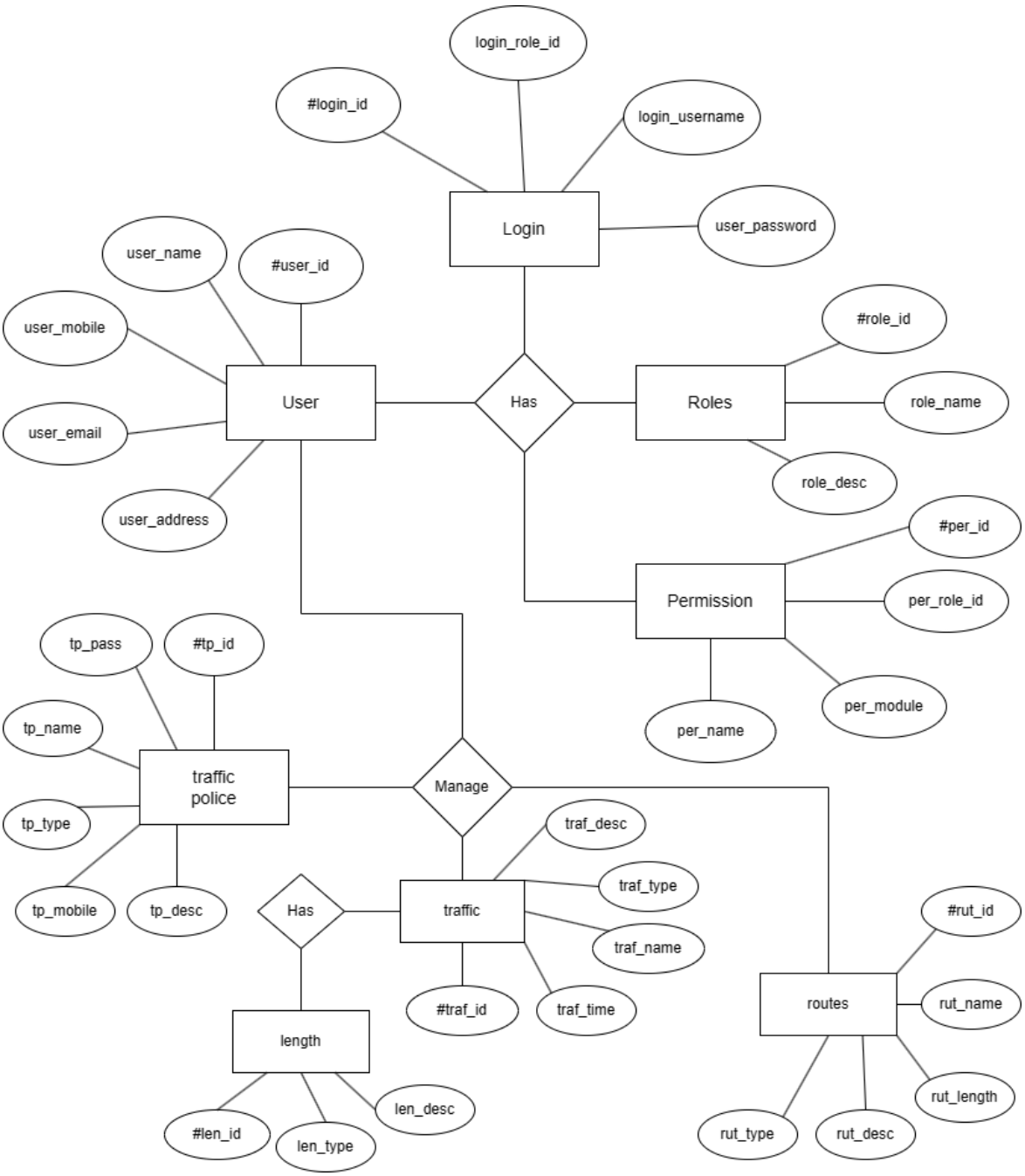

Figure 3: ER Diagram (Group Study)

3.2.3 System Flow Diagram

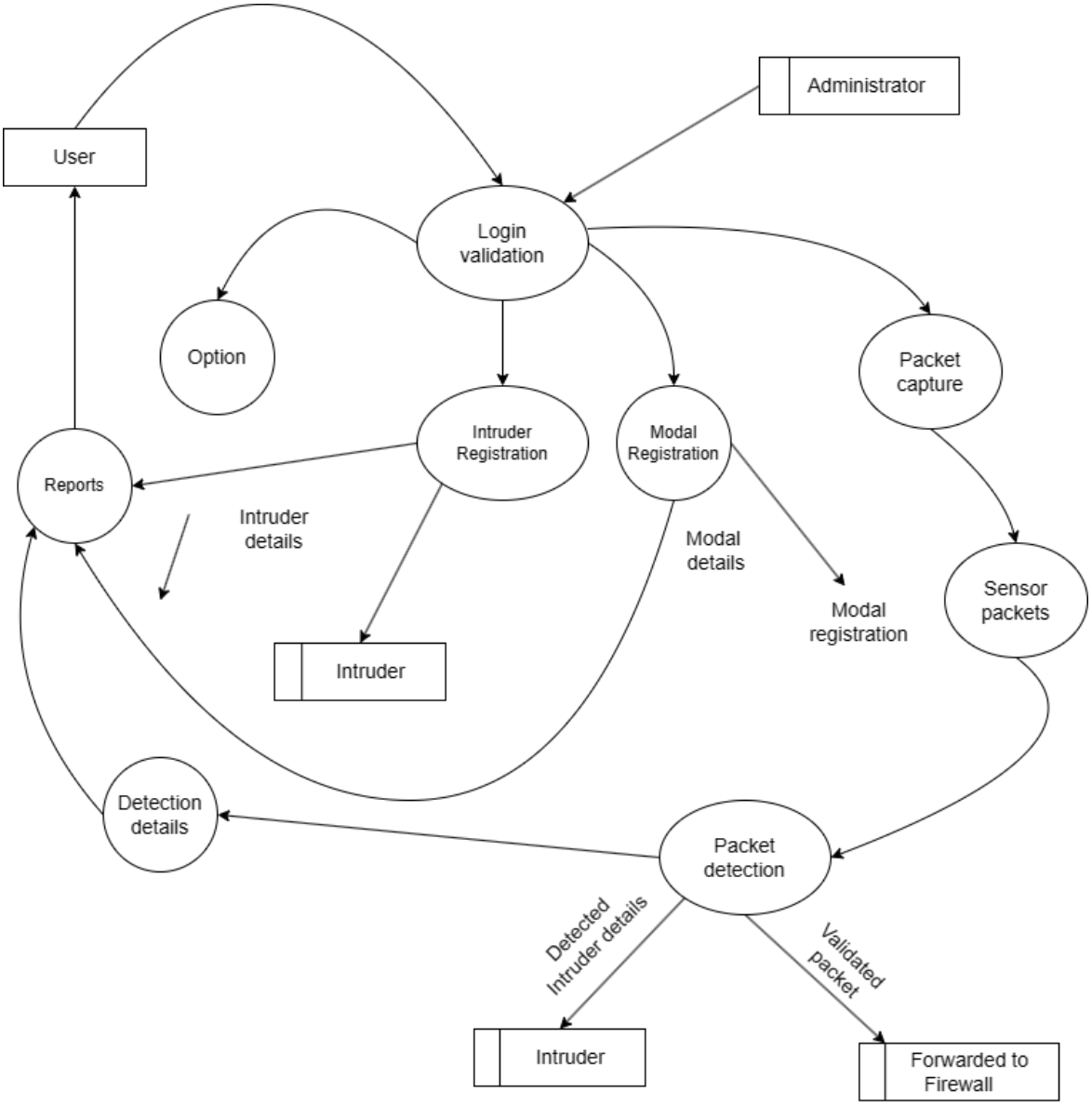

Figure 4 : System Flow Diagram (Group Study)

3.2.4 Use Case Diagram

A use case diagram is a visual representation that illustrates the interactions between actors and use cases within a system, commonly used in software development to capture functional requirements.

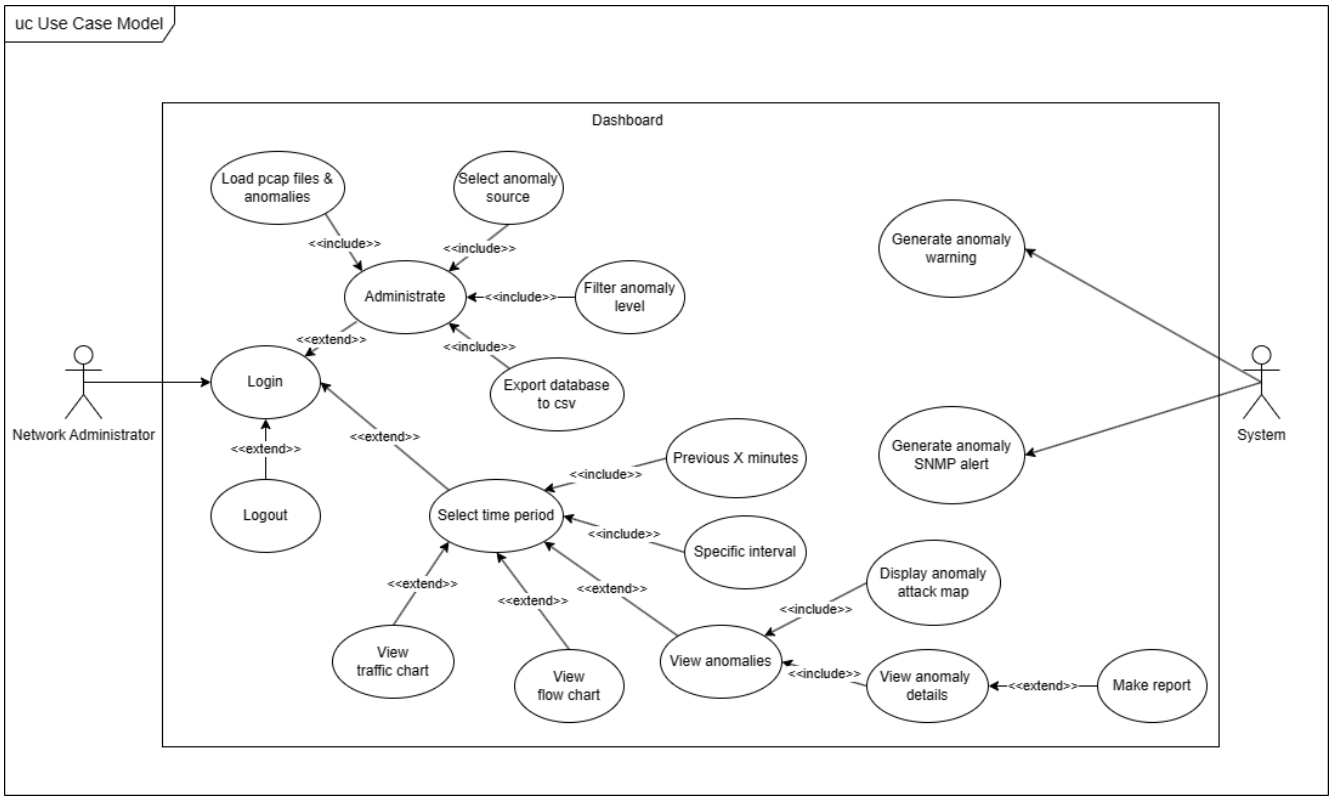

Figure 5: Use Case Diagram (Group Study)

3.2.5 System Flow Chart

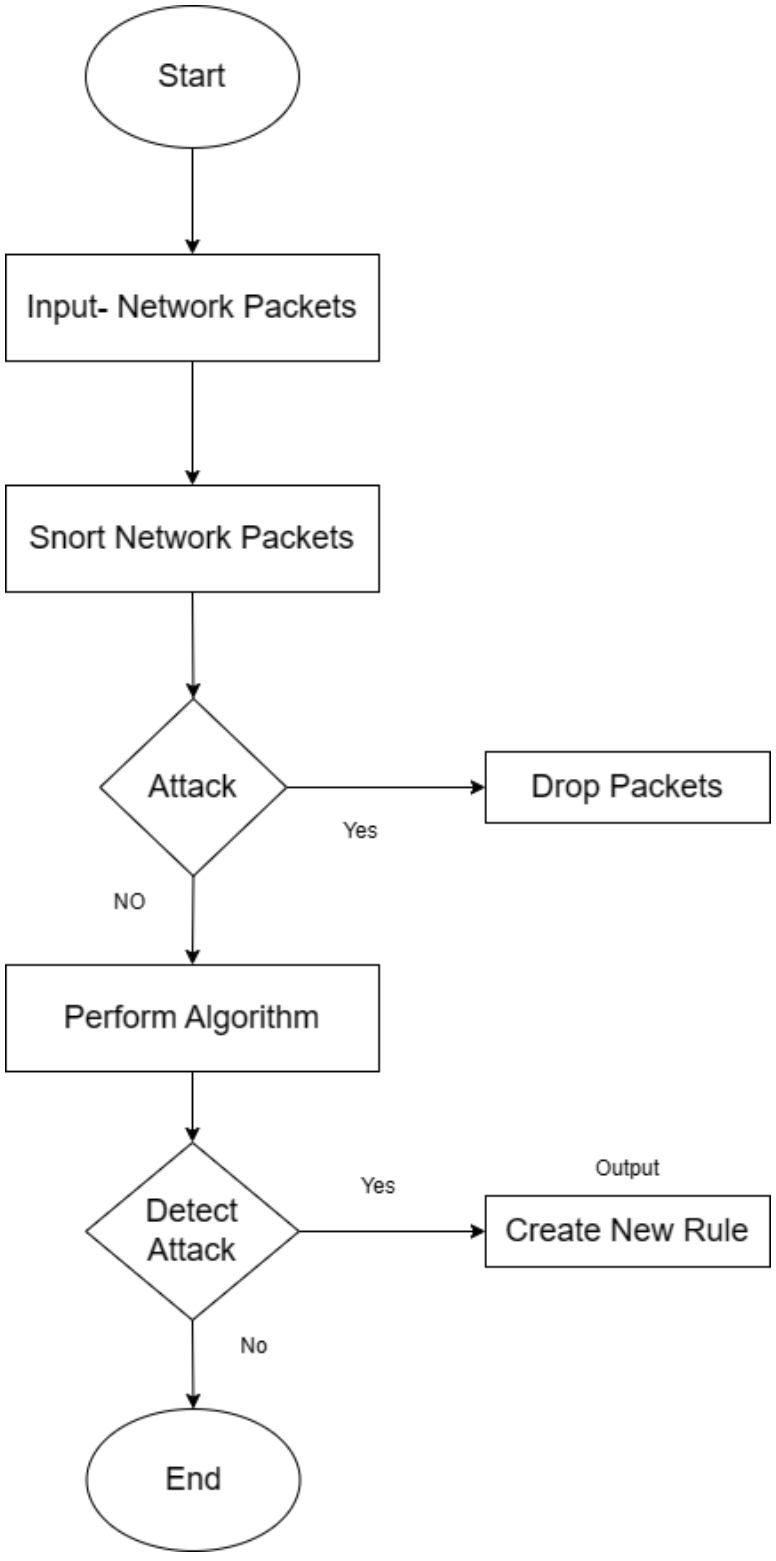

Figure 6: Flow Chart of NIDS (Group Study)

3.3 Software Development and Hardware Required

Minimum Hardware Requirements:

Processor (CPU):

Quad-core processor or equivalent: Adequate processing power is essential for real-time packet analysis and anomaly detection.

Memory (RAM):

8 GB or more: Sufficient RAM is crucial for efficiently storing and processing data during network monitoring.

Storage:

100 GB or more: Adequate storage space is needed for log files, captured packets, and any database used for storing detection data.

Network Interface Cards (NICs):

Multiple Gigabit Ethernet NICs: To capture and analyze network traffic effectively, having multiple NICs can be beneficial, especially if deploying on a high-traffic network.

Minimum Software Requirements:

Operating System:

Linux distribution (e.g., Ubuntu, CentOS, Debian): Linux is commonly preferred for NIDS implementations due to its stability and support for open-source tools.

Packet Capture and Analysis:

Wireshark or tcpdump: For capturing and analyzing network packets.

Scapy: Python library for crafting and analyzing packets.

Intrusion Detection Software:

Snort or Suricata: Open-source intrusion detection systems with signature-based and anomaly-based detection capabilities.

3.4 Testing and Maintenance

For proper implementation of this web application. Application should be tested at developer side to ensure everything work properly. We will do manual testing of our project and include our testing report after the completion of final project.

3.5 Algorithm Selection

In this section, elaborate on the use of the Random Forest classifier for intrusion detection:

3.5 Algorithm Selection: Random Forest Classifier

Our choice for the machine learning algorithm to classify network traffic for intrusion detection is the Random Forest classifier. Random Forest is an ensemble learning method that combines multiple decision trees to enhance the accuracy and robustness of the model. Its ability to handle high-dimensional data and capture complex relationships makes it well-suited for the dynamic and evolving nature of network intrusion patterns [23].

Random Forest operates by constructing a multitude of decision trees during training and outputs the mode of the classes for classification tasks. Each decision tree in the ensemble is trained on a subset of the dataset, and the randomness introduced in the process helps prevent overfitting and enhances the generalization capability of the model [23]. If we find some problem with Random Forest Classifier Algorithm during the process regarding proper accurate data output, we will be using KNN algorithm or CNN for it because looking through the case studies and some project in GitHub these technologies have been successfully used to detect the anomalies with appreciable accuracy but at end we will be using that algorithm which is gives proper output required with our data.

3.5.1 Training Methodology

Detail the process of training the Random Forest classifier for intrusion detection:

3.5.1 Training Methodology: Random Forest Classifier

To train the Random Forest classifier for intrusion detection, we follow a systematic approach:

1. **Data Preparation:**
    - **Dataset Splitting:** The collected network traffic data is divided into training and testing sets. A significant portion is allocated for training to ensure the model learns effectively.
2. **Feature Extraction:**
    - **Selection of Features:** Relevant features are extracted from the preprocessed data, including packet attributes, protocol information, and other relevant metadata. Feature importance analysis may guide the selection process.
3. **Training the Random Forest Model:**
    - **Ensemble Construction:** Multiple decision trees are trained on different subsets of the training data, introducing variability into the model.
    - **Hyperparameter Tuning:** Parameters such as the number of trees, maximum depth, and minimum samples per leaf are optimized through cross-validation to enhance model performance.
4. **Validation and Testing:**
    - **Model Validation:** The trained Random Forest model is validated using the testing dataset to assess its performance and generalization capabilities.
    - **Evaluation Metrics:** Performance metrics such as precision, recall, F1-score, and accuracy are calculated to quantify the effectiveness of the model in detecting intrusions.

3.5.2 Evaluation Metrics

Specify the evaluation metrics used to assess the performance of the Random Forest classifier:

3.5.2 Evaluation Metrics: Random Forest Classifier

The performance of the Random Forest classifier will be evaluated using the following metrics:

- **Accuracy:** Measures the overall correctness of the classification.

- **Precision:** Quantifies the proportion of true positive predictions among all instances predicted as positive.
- **Recall (Sensitivity):** Measures the proportion of actual positive instances correctly predicted by the model.
- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure of a model's performance.

These metrics will provide insights into the effectiveness of the Random Forest classifier in distinguishing between normal network activities and potential intrusions.

Chapter 4: Time Estimation

Before getting started with any work, proper preparation of working schedule containing of all the task that should be done for completion of project is necessary. For this purpose, we have included Gantt chart with time estimation in a total span of 3-4 months is included below.

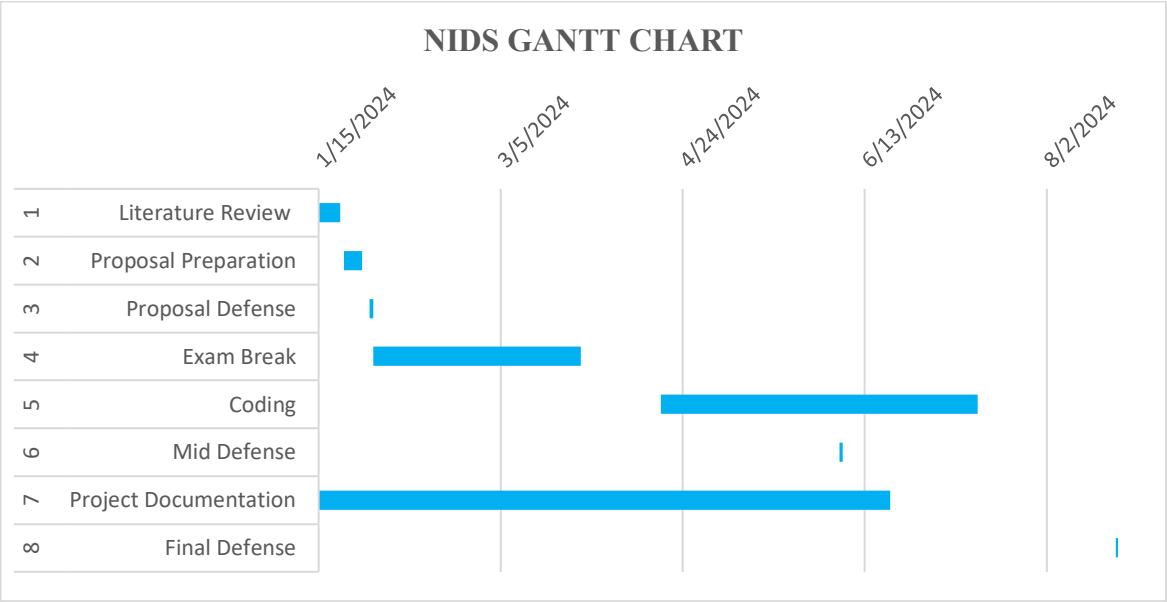

Figure 7: Gantt Chart (Group Study)

Chapter 5: Epilogue

5.1 Expected Outcome

After completion of this project, we have expected to fulfill all the objective of this Network Intrusion Detection System with features to scan the network traffic and handle the issues efficiently with help of this technology which will eventually be helpful in this modern era of information and technology.

5.2 Budget Analysis

Total cost for development of this project is mentioned below:

Name	Cost	Quantity	Sub Total
Printing	250	3	750
Human Resources	3000	4	12000
Internet Package	100	20	2000
Total Cost			NRs.14750

References

[1] J. Fox, "Cobalt," Cobalt, 7 Oct 2022. [Online]. Available: https://www.cobalt.io/blog/biggest-cybersecurity-attacks-in-history. [Accessed 22 Jan 2024].

[2] C. D. Lirim Ashiku, "Network Intrusion Detection System using Deep Learning," Procedia Computer Science , Malvern, Pennsylvania, 2021.

[3] D. B. M. Jabez, "Intrusion Detection System," Procedia Computer Science, Bhubaneswar, Odhisha, India, Intrusion Detection System (IDS).

[4] H. A. M. Jimmy Shun, "Network Intrusion Detection System Using Neural Network," in *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, 2020.

[5] W. P. &. R. A. Nasrin Sultana. Naveen Chilamkurti, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer to Peer Networking and Applications,* vol. 12, no. https://doi.org/10.1007/s12083-017-0630-0, pp. 493-501, 2019.

[6] R. S. Y. G. S. V. T. Shanbhag, "A high-performance network intrusion detection system," 1999.

[7] T. N. L. L. D. E. O. D. O. B. L. M. R. Patrick Vanin, "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," *Applied Sciences,* vol. 12, no. https://doi.org/10.3390/app122211752, p. 22, 2022.

[8] C. Staff, "Coursera.org," Coursera, 21 Nov 2023. [Online]. Available: https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python. [Accessed 22 1 2024].

[9] C. BasuMallick, "SpiceWorks," 20 Mar 2023. [Online]. Available: https://www.spiceworks.com/tech/devops/articles/c-vs-cplus/. [Accessed 22 1 2024].

[10] R. Juillet, "BOCASAY_," 19 Apr 2022. [Online]. Available: https://www.bocasay.com/what-you-need-know-about-java-programming-language/. [Accessed 22 1 2024].

[11] R. R. M. M. S. G. Rohith Raj S, "SCAPY - A powerful interactive packet manipulation program," *IEEE,* no. https://doi.org/10.1109/ICNEWS.2018.8903954, p. 14, 2018.

[12] J. Breeden, "Networkworld," 08 Jun 2022. [Online]. Available: https://www.networkworld.com/article/971145/what-is-wireshark.html. [Accessed 22 1 2024].

[13] K. ,. G. A. S. S. Neha V Sharma, "RETRACTED: Performance Study of Snort and Suricata for Intrusion Detection System," in *IOP Conference Series: Materials Science and Engineering*, 2021.

[14] Ł. Ruczyński, "netguru," 8 Dec 2022. [Online]. Available: https://www.netguru.com/blog/top-machine-learning-frameworks-compared. [Accessed 22 Jan 2024].

[15] S. Ravoof, "Kinsta," 29 Dec 2023. [Online]. Available: https://kinsta.com/blog/postgresql-vs-mysql/. [Accessed 22 Jan 2024].

[16] J. Gopalakrishnan, "Knowi," [Online]. Available: https://www.knowi.com/blog/what-is-elastic-search/. [Accessed 22 Jan 2024].

[17] S. Das, "Browser Stack," 14 Mar 2023. [Online]. Available: https://www.browserstack.com/guide/top-python-web-development-frameworks. [Accessed 22 Jan 2024].

[18] Simplilearn, "Simplilearn," 16 Jan 2024. [Online]. Available: https://www.simplilearn.com/data-visualization-tools-article. [Accessed 22 Jan 2024].

[19] R. Dickens, "Encryption Consulting," 12 May 2021. [Online]. Available: https://www.encryptionconsulting.com/what-are-encryption-protocols-and-how-do-they-work/. [Accessed 22 Jan 2024].

[20] F. Kane, "Coralogix," 12 Jan 2021. [Online]. Available: https://coralogix.com/blog/a-practical-guide-to-logstash-syslog-deep-dive/. [Accessed 22 Jan 2024].

[21] R. J. Tara Salman, "NETWORKING PROTOCOLS AND STANDARDS FOR INTERNET OF THINGS," vol. 1, no. 9781119173601, pp. 215-238, 2017.

[22] S. H. Perveez, "SimpliLearn," 27 Jul 2023. [Online]. Available: https://www.simplilearn.com/tutorials/git-tutorial/what-is-git. [Accessed 22 Jan 2024].

[23] D. L. B. Abebe Tesfahun, "Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction," *IEEE,* no. 10.1109/CUBE.2013.31., pp. 127-132, 2013.

Certificate of Project Proposal Approval

This project entitle “Network Intrusion Detection System” proposed by the students Prajjwal Adhikari, Raj Gurung, Rozal Dahal and Aman Paudel of United Technical College, under the department of Computer Engineering has been submitted as per the content, Style and format proposed by research and development. The project has been feasible and thus has been approved.

……..

Department Head of Computer Engineering,

Er. Sahit Baral

United Technical College, Bharatpur-11

Chitwan