# SDN-BASED INTELLIGENT INTRUSION DETECTION SYSTEM (IIDS) USING MACHINE LEARNING

Group ID: RP-24-25J-120

**Research Logbook**

**Satkurulingam.S**

**IT21282072**

**BSc (Hons) Degree in Information Technology Specialized in Cyber Security**

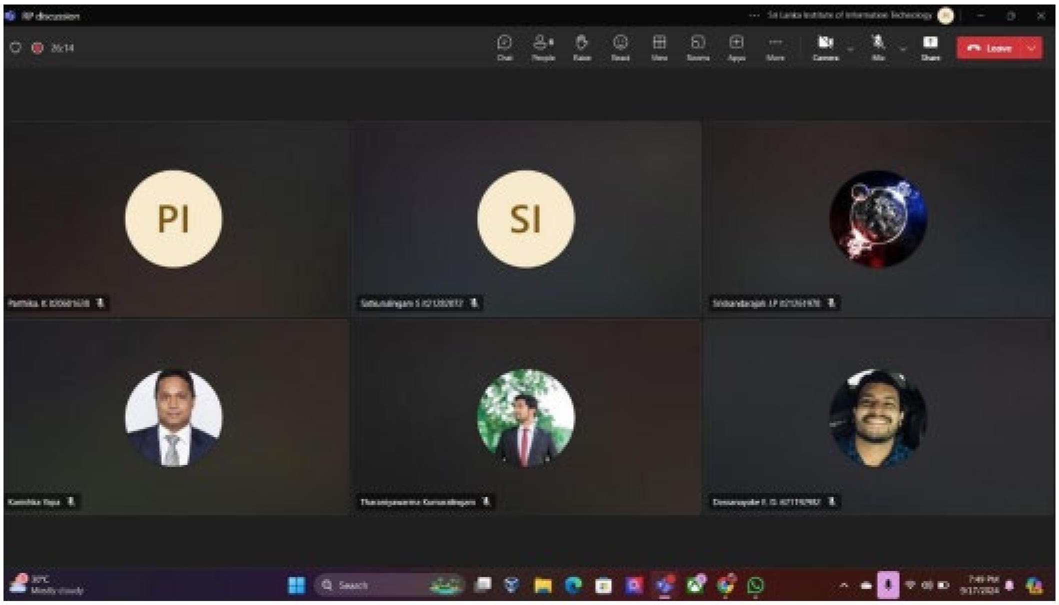
**Sri Lanka Institute of Information Technology Sri Lanka**

**June 2025**

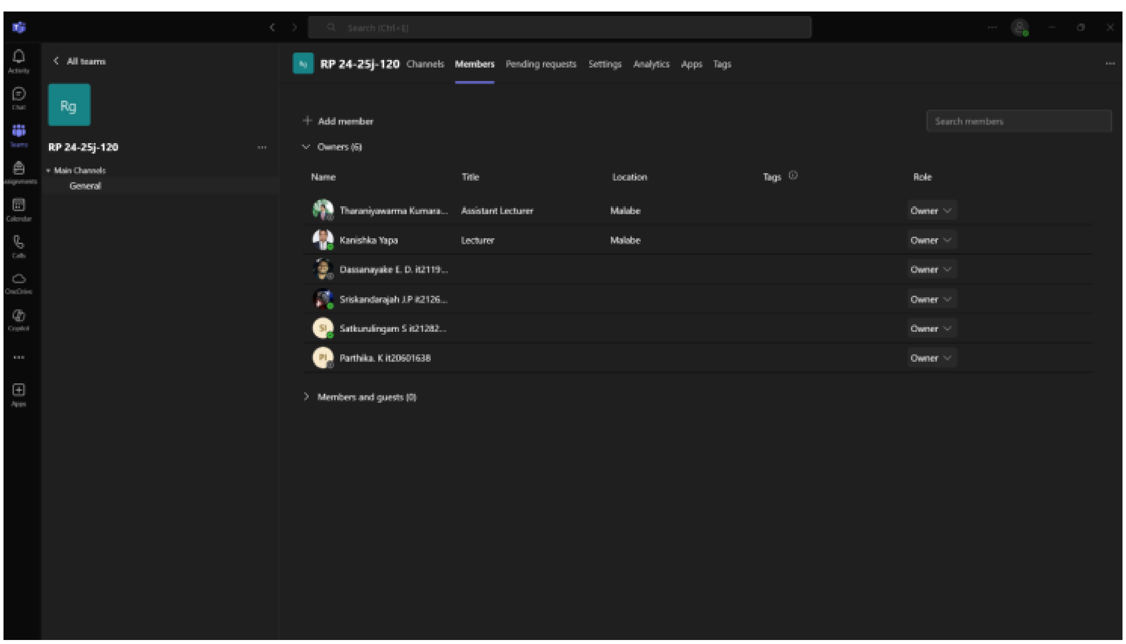
## Tasks

**Meeting with the supervisor to discuss the project topic for the first time.**

- Physically meet the supervisor.
- Discuss the research project topic area.
- Get the supervisor's ideas about the research topic via Microsoft Teams.

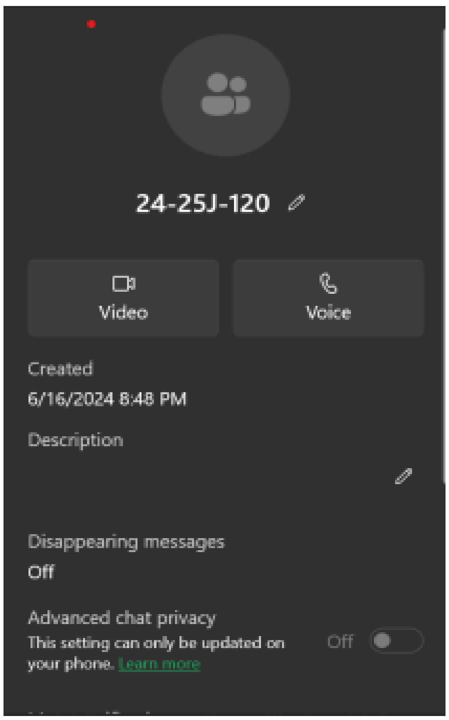


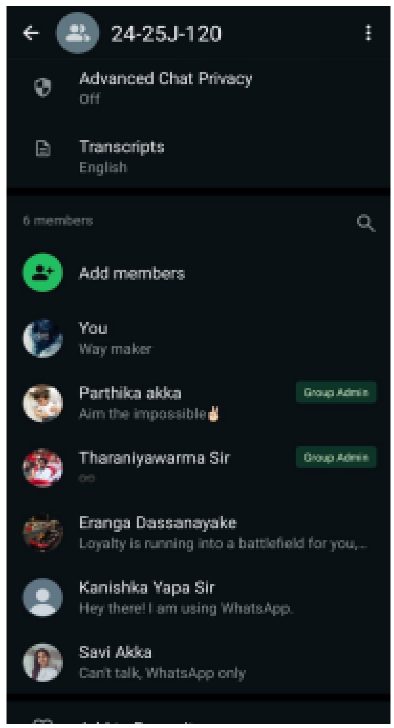

**Created separate MS Teams channels for Conversation**



**Created the Research Team WhatsApp Group.**

- Discuss the research topic with team members.
- Discuss the research problem.
- Get the solution ideas with brainstorming sessions.
- Identify the main solutions.
- Assign tasks and conversation highlights.



## Completed Task and Conversation highlights.

• Creating the proposal document at supervisor request.

• Doing a literature review upon supervisor request

In the dynamic and rapidly evolving landscape of cybersecurity, traditional network architectures are increasingly challenged by sophisticated and persistent threats. Software-Defined Networking (SDN) emerges as a transformative technology that offers centralized control and flexibility in network management. However, this very centralization introduces new vulnerabilities that adversaries can exploit, particularly in the form of flow rule manipulation, IP spoofing, Distributed Denial of Service (DDoS) attacks, and SQL injection attacks. The proposed research seeks to address these vulnerabilities by developing an SDN-based Intelligent Intrusion Detection System (IIDS) powered by Machine Learning (ML) to enhance network security.

Flow rule manipulation is a critical threat in SDN environments. Attackers can alter or inject malicious flow rules, disrupting network traffic and compromising data integrity. Traditional intrusion detection mechanisms often fail to detect such sophisticated attacks due to their reliance on predefined signatures or static rule sets. The first component of this research focuses on developing a Machine Learning-based Intrusion Detection System (IDS) specifically for detecting flow rule manipulation. By leveraging ML algorithms, this component aims to analyze network traffic patterns, identify anomalies indicative of flow rule tampering, and trigger appropriate defensive actions, thus safeguarding the integrity and reliability of the SDN infrastructure.

IP spoofing is a deceptive technique where an attacker impersonates a legitimate IP address to gain unauthorized access to network resources or launch further attacks, such as Man-in-the-Middle (MitM) or Denial of Service (DoS). Traditional security measures often struggle to accurately detect and prevent IP spoofing due to its dynamic nature. The second component of this research will develop a Machine Learning-based IDS designed to detect IP spoofing attacks in real time. By analyzing packet headers, flow data, and behavioral patterns, the system will identify discrepancies between the source IP address and the actual origin of the traffic. This will enable the network to swiftly isolate and mitigate IP spoofing attempts, thereby enhancing network resilience.

## Proposed Machine Learning based Intrusion Detection Engine System

| | | | |
|---|---|---|---|
| Satkurulingam.S | Develop a Machine Learning Based Intrusion Detection System to IP spoofing attack | **1. Problem Definition and Research**<br><br>Define the scope of the IDS and research IP spoofing attack vectors.<br><br>Identify the dataset required for training and testing (e.g., network traffic data).<br><br>**2. Data Collection** | 1. **Dynamic Learning and Adaptation**: The IDS uses machine learning algorithms to dynamically learn from network traffic patterns, adapting to new and evolving IP spoofing attack strategies, unlike static, rule-based systems. |

| | | | |
|---|---|---|---|
| | | Collect or generate a dataset that includes both normal and spoofed IP traffic.<br><br>Ensure the dataset is labeled for supervised learning.<br><br>**3. Data Preprocessing**<br><br>Clean and normalize the data (remove noise, handle missing values).<br><br>Extract relevant features (e.g., packet headers, traffic patterns).<br><br>Split the data into training and testing sets.<br><br>5. **Feature Engineering** Perform feature selection or extraction to identify the most important features that differentiate between normal and spoofed traffic.<br><br>**5. Model Selection** | **2. Real-Time Traffic Analysis:** The system incorporates real-time monitoring and analysis, enabling it to detect and respond to IP spoofing attacks as they occur, enhancing the responsiveness and accuracy of the IDS.<br><br>**3. Time-Series Analysis Integration:** By integrating time-series analysis, the IDS can detect temporal anomalies in network traffic, providing a deeper understanding of traffic behavior over time and identifying spoofing attempts that mimic legitimate traffic. |

| | | | |
|---|---|---|---|
| | | Choose appropriate machine learning algorithms (e.g., Random Forest, SVM, Neural Networks).<br><br>Consider using ensemble methods for better performance.<br><br>**6. Model Training**<br><br>Train the chosen models on the preprocessed dataset.<br><br>Use cross-validation to tune hyperparameters and prevent overfitting.<br><br>**7. Model Evaluation**<br><br>Evaluate the model using metrics like accuracy, precision, recall, and F1-score.<br><br>Perform testing on the separate test dataset to assess generalization.<br><br>**8. Model Optimization** | 4. **Ensemble Learning Techniques**: The system employs ensemble learning, combining algorithms like Random Forest and Neural Networks, to improve detection accuracy and robustness across diverse network environments.<br><br>5. **Specialized Feature Engineering**: The IDS focuses on extracting and selecting features specifically relevant to IP spoofing, such as anomalies in packet headers and irregular IP address patterns, enhancing its ability to detect spoofed traffic. |

| | | | |
|---|---|---|---|
| | | Fine-tune the model based on evaluation results.<br><br>Implement techniques like feature scaling, regularization, or additional data augmentation if necessary.<br><br>**9. Implementation**<br><br>Integrate the trained model into an IDS framework.<br><br>Develop real-time monitoring capabilities to detect IP spoofing in live traffic.<br><br>**10. Testing and Deployment**<br><br>Conduct extensive testing in a simulated or real network environment.<br><br>Deploy the IDS in a production environment and monitor its performance. | 6. **Continuous Model Updates:** With access to real-time data, the IDS continuously updates its models, ensuring it remains effective against emerging and sophisticated IP spoofing threats.<br><br>7. **Resilience Against Evasion Techniques:** The system is designed to detect subtle anomalies that may evade traditional detection methods, offering enhanced protection against advanced spoofing techniques. |

In the dynamic and rapidly evolving landscape of cybersecurity, traditional network architectures are increasingly challenged by sophisticated and persistent threats. Software-Defined Networking (SDN) emerges as a transformative technology that offers centralized control and flexibility in network management. However, this very centralization introduces new vulnerabilities that adversaries can exploit, particularly in the form of flow rule manipulation, IP spoofing, Distributed Denial of Service (DDoS) attacks, and SQL injection attacks. The proposed research seeks to address these vulnerabilities by developing an SDN-based Intelligent Intrusion Detection System (IIDS) powered by Machine Learning (ML) to enhance network security.

Flow rule manipulation is a critical threat in SDN environments. Attackers can alter or inject malicious flow rules, disrupting network traffic and compromising data integrity. Traditional intrusion detection mechanisms often fail to detect such sophisticated attacks due to their reliance on predefined signatures or static rule sets. The first component of this research focuses on developing a Machine Learning-based Intrusion Detection System (IDS) specifically for detecting flow rule manipulation. By leveraging ML algorithms, this component aims to analyze network traffic patterns, identify anomalies indicative of flow rule tampering, and trigger appropriate defensive actions, thus safeguarding the integrity and reliability of the SDN infrastructure.

IP spoofing is a deceptive technique where an attacker impersonates a legitimate IP address to gain unauthorized access to network resources or launch further attacks, such as Man-in-the-Middle (MitM) or Denial of Service (DoS). Traditional security measures often struggle to accurately detect and prevent IP spoofing due to its dynamic nature. The second component of this research will develop a Machine Learning-based IDS designed to detect IP spoofing attacks in real time. By analyzing packet headers, flow data, and behavioral patterns, the system will identify discrepancies between the source IP address and the actual origin of the traffic. This will enable the network to swiftly isolate and mitigate IP spoofing attempts, thereby enhancing network resilience.

DDoS attacks continue to be one of the most disruptive and widespread threats in the digital landscape, capable of overwhelming network resources and rendering services unavailable. SDN's centralized architecture, while offering numerous advantages, also presents a single point of failure that attackers can target with DDoS attacks. The third component of this research will focus on creating a Machine Learning-based Intrusion Detection Engine to identify and counteract DDoS attacks. By monitoring traffic flow patterns and identifying unusual surges or irregularities, the system will differentiate between legitimate traffic spikes and malicious DDoS attempts, ensuring timely and effective responses to protect network availability.

SQL injection attacks are a prevalent threat to databases, where attackers exploit vulnerabilities in SQL queries to manipulate or access unauthorized data. In an SDN context, such attacks can compromise critical network databases, leading to severe consequences. The fourth component of this research aims to develop a Machine Learning-based Intrusion Detection Engine that can detect SQL injection attacks. By analyzing query patterns and database interactions, the system will identify potential injection attempts, thereby preventing unauthorized database access and maintaining the integrity of network operations.

Developing a Machine Learning-based Intrusion Detection System (IDS) for IP spoofing attacks requires a blend of specialized domain expertise and specific data requirements. Expertise in cybersecurity and networking is essential to understand the nuances of TCP/IP protocols and how IP spoofing exploits these systems. This knowledge helps identify critical patterns in network traffic that signify an attack. Additionally, proficiency in machine learning, particularly in supervised learning algorithms like Random Forest, SVM, and Neural Networks, is crucial for developing models that can accurately distinguish between legitimate and malicious traffic. Data science skills are needed to manage, preprocess, and analyze large network traffic datasets, including tasks like feature extraction and time-series analysis. The data requirements include a labeled dataset containing both normal and spoofed IP traffic, rich in features such as IP addresses and packet headers, which are critical for accurate detection. Access to real-time network data is also necessary for continuous training and evaluation, allowing the IDS to adapt to evolving attack patterns and maintain high detection accuracy.

## Completed Task and Conversation Highlights.

- Determining the components for each member and discussing with the Supervisor.
- Fine tuning the scope for each component.
- Discussing the proposed components with co-supervisor.
- Find the Related research paper for individual SDN Component.
- Get a full idea of each research paper.
- Mark down the not covering SDN areas in these research papers.
- Identify the novelty parts of each individual component.
- Creating the Topic Assignment Form (TAF)
- Getting the approval from the Supervisor.

### Defense Against Software-Defined Network Topology Poisoning Attacks

Yang Gao and Mingdi Xu*

**Abstract:** Software-Defined Network (SDN) represents a new network paradigm. Unlike conventional networks, SDNs separate control planes and data planes. The function of a data plane is enabled using switches, whereas that of a control plane is facilitated by a controller. The controller learns network topologies and makes traffic forwarding decisions. However, some serious vulnerabilities are gradually exposed in the topology management services of current SDN controller designs. These vulnerabilities mainly exist in host tracking and link discovery services. Attackers can exploit these weak points to poison the network topology information in SDN controllers. In this study, a novel solution is proposed to defend against topology poisoning attacks. By analyzing the existing topology attack principles and threat models, this work constructs legal conditions for host migration to detect host hijacking attacks. The checking of the Link Layer Discovery Protocol (LLDP) source and integrity is designed to defend against link fabrication attacks. A relay-type link fabrication attack detection method based on entropy is also designed. Results show that the proposed solution can effectively detect existing topological attacks and provide complete and comprehensive topological security protection.

**Key words:** Software-Defined Network (SDN); topology discovery; topology poisoning attacks

**1 Introduction**

Software-defined networks (SDNs), originating from the campus network of Stanford University, were proposed to solve the bloated and inefficient problems of traditional networks. Through the separation of the data forwarding and routing control of the traditional Internet in SDNs, the centralized control and distributed forwarding of these networks can be realized. The form of programming provides an interface to the outside world[1, 2]. The dynamic and flexible characteristics of SDNs have attracted widespread attention from academia and industries. Researchers in many fields actively use SDNs to build new systems for solving problems, such as the limited scalability of traditional architectures, wireless sensor networks[3], the Internet of Things[4], and optical networks[5]. The recent studies on the security of the SDN topology discovery mechanism mainly involve three aspects: design of a security framework[6–9], construction and addition of a new protocol[10–12], and encryption authentication[13]. TopoGuard[14] is a security extension of the SDN controller, and it detects attacks on the SDN network topology view by fixing the security vulnerabilities in the controller. However, TopoGuard is unable to detect switch-based link fabrication attacks. PolicyTopo[9] proposes a solution to determine the link status on the basis of the information entropy of the network delay. Topology attacks are distinguished by the threshold when the network delay is low. When the network delay is high, PolicyTopo detects attacks by secure ports. The disadvantage is that the state only depends on the linear relationship of adjacent entropy values and the definition of secure ports brings further burden to the network. Reference [8] proposed a defense scheme on the basis of a statistical analysis of link delays to detect

• Yang Gao and Mingdi Xu are with the Platform Research and Development Department, Wuhan Institute of Digital Engineering, Wuhan 430073, China. E-mail: Shirley_Stefani@163.com; mingdixu@163.com.
* To whom correspondence should be addressed.

### Topology Poisoning Attacks and Prevention in Hybrid Software-Defined Networks

Pragati Shrivastava*, Kotaro Kataoka†
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Email: *cs14resch11007@iith.ac.in, †kotaro@cse.iith.ac.in

***Abstract*—The hybrid software-defined networks (SDN) architectures are beneficial for a smooth transition and less costly SDN deployment. However, legacy switches and SDN switches coexistence brings new challenges of deployment inconsistency management and security. Security is not well studied for hybrid SDN architecture. In this paper, we study the topology poisoning attacks in hybrid SDN for the first time. We propose new attack vectors for link fabrication in hybrid SDN. The new attack is named "multi-hop link fabrication", in which an adversary successfully injects a fake multi-hop link (MHL) by exploiting the link discovery protocols. We presented the Hybrid-Shield, a link verification framework for hybrid SDN link discovery. Hybrid-Shield introduces a novel verification technique that includes: i) monitoring legacy switch and host generated traffic at MHL and ii) validating the existence of legacy switches contained in an MHL. This paper presents the prototype implementation of Hybrid-Shield over a real SDN controller. The experimental evaluation is performed with the mininet virtual network emulation. Our evaluation shows that Hybrid-Shield is capable of detecting MHL fabrication attacks in real-time with high accuracy. Hybrid-Shield's performance evaluation shows that it is lightweight at the controller as it causes less overhead and requires no additional functionalities at the SDN controller for deployment.**

***Index Terms*—Software-defined Networks (SDN), Topology Poisoning , OpenFlow, Hybrid SDN**

I. INTRODUCTION

debugger Anteater [7] require further consideration of the hybrid SDN deployment.

The SDN controller exchanges the control messages with the SDN switches through the southbound protocol (e.g., OpenFlow) and collects the information about network device connectivity, flow statistics, and traffic. This information collectively provides the abstract view of the network to SDN applications and is utilized for forwarding decisions. An attacker can exploit the communication between the controller and switches and damage the controller's network view. This tampered network state causes dysfunction of the network applications, such as QoS and Load balancing. These attacks are called Topology Poisoning attacks. Topology Poisoning attacks can trigger more severe attacks like Denial-of-Service (DoS) at the controller. Protection from Topology Poisoning attacks become more challenging in hybrid SDN deployment than the pure SDN. The fundamental challenges for defending against topology poisoning attacks in a hybrid SDN are: 1) The controller cannot control the legacy part of the network, i.e., the controller cannot instruct legacy switches in hybrid SDN. 2) The legacy part of the network is unknown to the controller, i.e., the controller does not have the complete topology view in hybrid SDN. 3) The controller cannot directly monitor the topology changes in the legacy network. The controller can

SLIIT UNI
THE KNOWLEDGE UNIVERSITY

**IT4010 – Research Project - 2024**

**Topic Assessment Form**

**Project ID :** 24-25J-120

1. Topic (12 words max)

SDN-based Intelligent Intrusion Detection System (IIDS) using Machine Learning

2. Research group the project belongs to

**Computing Infrastructure and Security (CIS)**

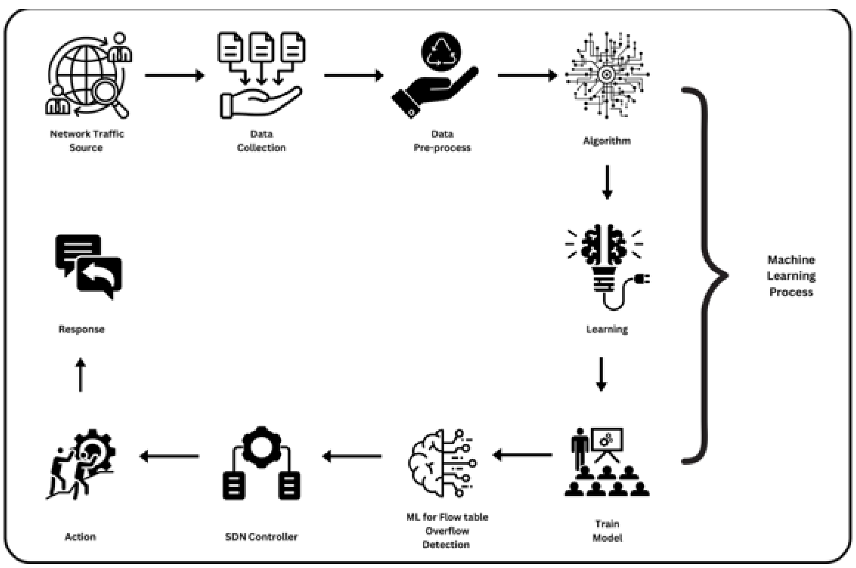
3. Research area the project belongs to

**Cyber Security (CS)**

4. If a continuation of a previous project:

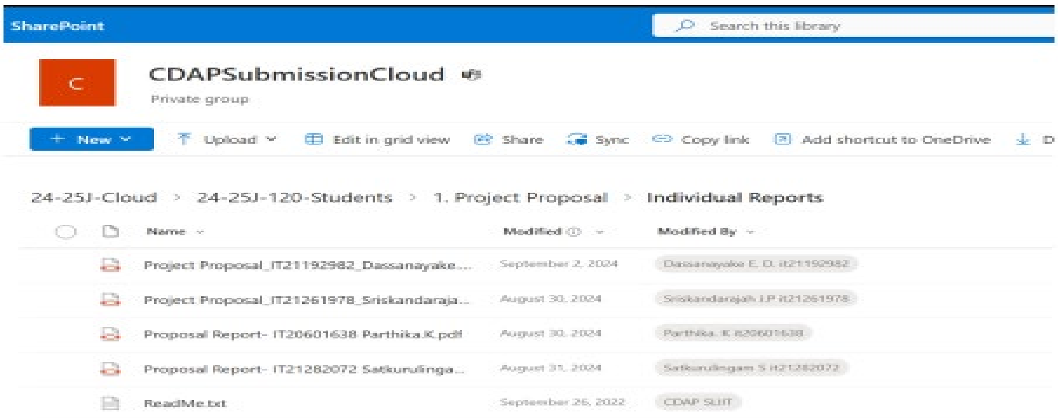
## Complete Task and Conversation Highlights

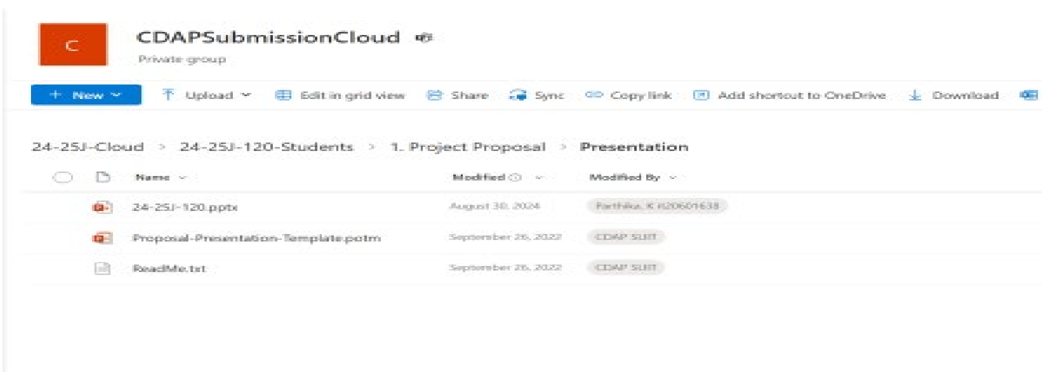
- Dividing the software components.
- Doing a thorough background investigation on each component.
- Creating the system architecture diagram of the proposed system.
- Discussing architecture with the Supervisor and Co-supervisor physically Meeting.

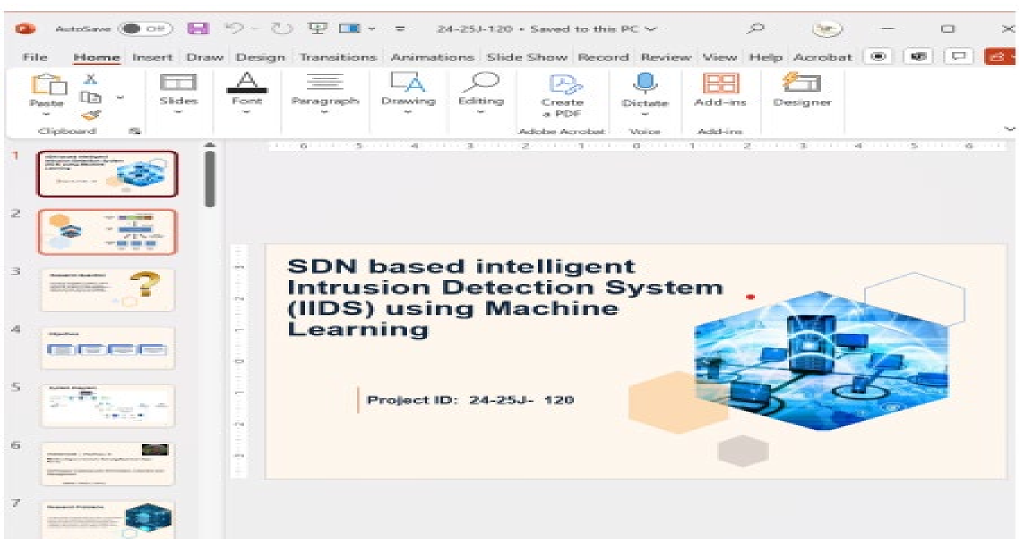


**Complete Task and Conversation Highlights.**

- Finalizing the components and getting ready for the progress presentation.
- Discussion the project with the supervisor before the proposal presentation.



### Completed Task and Conversation Highlights

• Finding the sample dataset until SDN system develop.

• Discussing with the co-supervisor the potential model and its accuracy and which model we should proceed with for the prediction.

clean_network_data
cleaned_topology_poisoning_dataset
generated_network_traffic_data

| | | | |
|---|---|---|---|
| topology_poisoning_dataset (1) | 12/29/2024 2:40 PM | Excel.CSV | 1 KB |
| topology_poisoning_dataset | 12/29/2024 2:35 PM | Excel.CSV | 1 KB |

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder, StandardScaler
X.loc[:, ['attack_impact', 'injected_metric']] = scaler.fit_transform(X[['attack_impact', 'injected_metric']])


# Load the dataset
df = pd.read_csv(r'C:\Users\hp\Downloads\topology_poisoning_attack_with_numeric_labels.csv')

# Preprocessing

# Encode categorical variables (node_type and link_status)
label_encoder = LabelEncoder()
df['node_type'] = label_encoder.fit_transform(df['node_type'])  # Benign=0, Malicious=1
df['link_status'] = label_encoder.fit_transform(df['link_status'])  # Up=1, Down=0

# Features (X) and Target (y)
X = df[['node_id', 'node_type', 'attack_impact', 'injected_metric', 'link_status']]
y = df['attack_status']  # 1 for attack, 0 for no attack

# Scale numerical features (optional, but helps many models)
scaler = StandardScaler()
X[['attack_impact', 'injected_metric']] = scaler.fit_transform(X[['attack_impact', 'injected_metric']])

# Split the dataset into training and testing sets
```

```
# Features (X) and Target (y)
X = df[['node_id', 'node_type', 'attack_impact', 'injected_metric', 'link_status']]
y = df['attack_status']  # 1 for attack, 0 for no attack

# Scale numerical features (optional, but helps many models)
scaler = StandardScaler()
X[['attack_impact', 'injected_metric']] = scaler.fit_transform(X[['attack_impact', 'injected_metric']])

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')

# Detailed classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

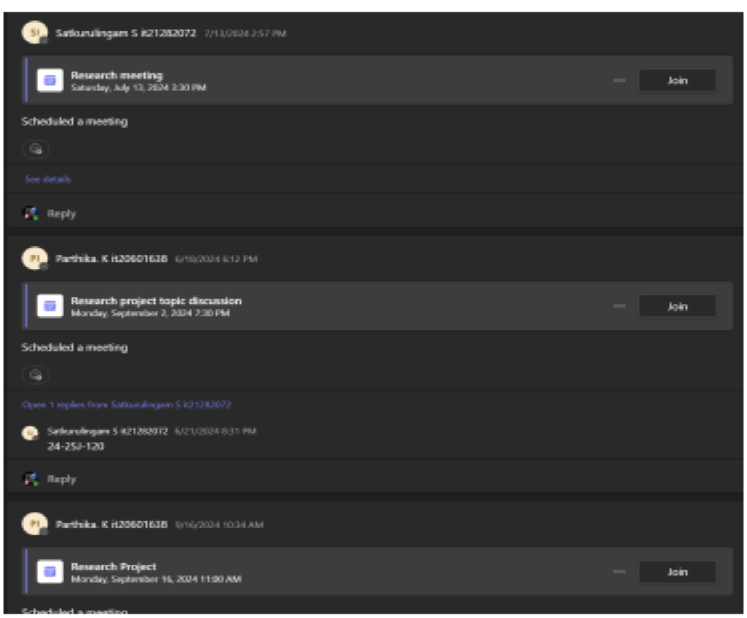
```
C:\Users\hp\AppData\Local\Temp\ipykernel_21192\1518277815.py:25: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X[['attack_impact', 'injected_metric']] = scaler.fit_transform(X[['attack_impact', 'injected_metric']])
Accuracy: 99.63%
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00      6192
           1       1.00      0.99      1.00      3808

    accuracy                           1.00     10000
   macro avg       1.00      1.00      1.00     10000
weighted avg       1.00      1.00      1.00     10000
```

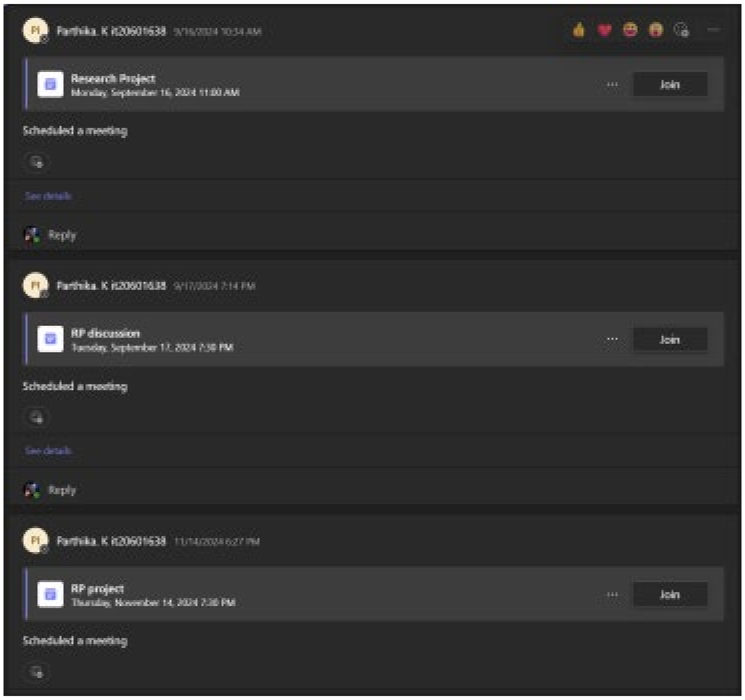
```python
from sklearn.model_selection import cross_val_score

# Perform cross-validation with Random Forest
scores = cross_val_score(model, X, y, cv=5)  # 5-fold cross-validation
print(f'Cross-validation scores: {scores}')
print(f'Mean accuracy: {scores.mean()}')
```

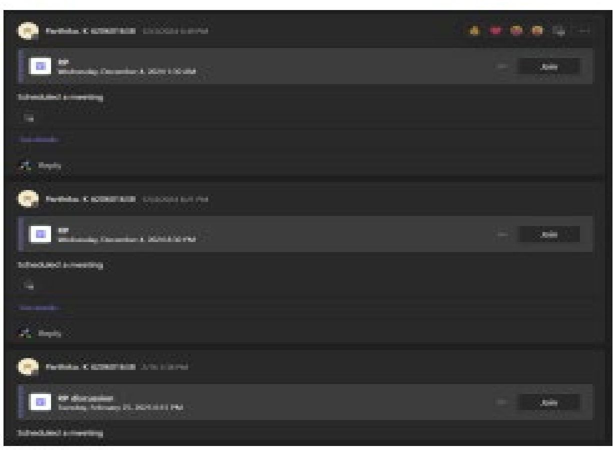
```
Cross-validation scores: [0.9969 0.9947 0.9952 0.9955 0.9961]
Mean accuracy: 0.9956800000000001
```

### Complete Tasks and Conversation Highlights

• Meeting with the research team and deciding the implementation milestone on Microsoft Teams.




Parthika. K it20601638 9/16/2024 10:34 AM
Research Project
Monday, September 16, 2024 11:00 AM
Join
Scheduled a meeting
See details
Reply
Parthika. K it20601638 9/17/2024 7:14 PM
RP discussion
Tuesday, September 17, 2024 7:30 PM
Join
Scheduled a meeting
See details
Reply
Parthika. K it20601638 11/14/2024 6:27 PM
RP project
Thursday, November 14, 2024 7:30 PM
Join
Scheduled a meeting


RP
Join
Scheduled a meeting
Reply
RP
Join
Scheduled a meeting
Reply
RP discussion
Join
Scheduled a meeting


PI
SI
Participants

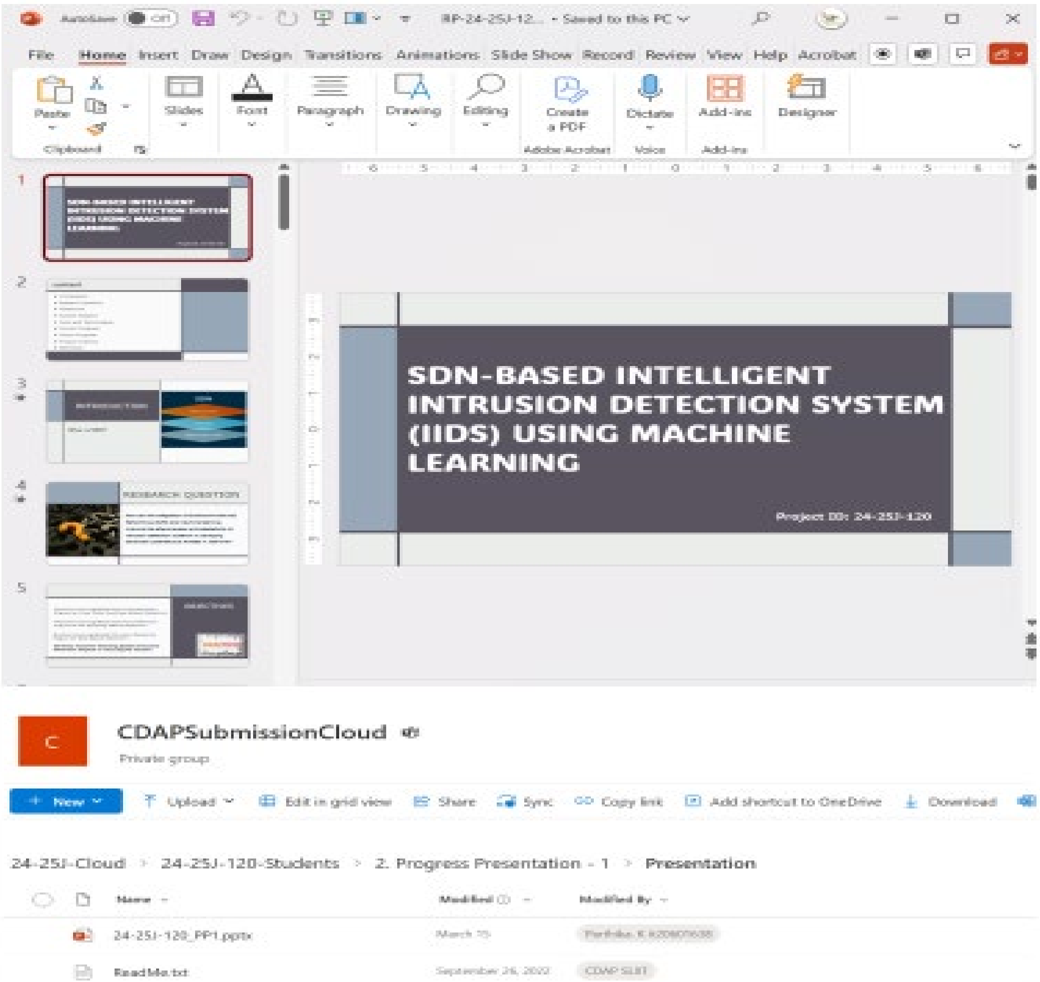
**Completed Tasks and Conversation Highlights**

• Prepare for Progress Presentation 1 (PP1).

• Creating the presentation.

• Finalizing the Projects.

• Communication with the supervisor after finalizing the project



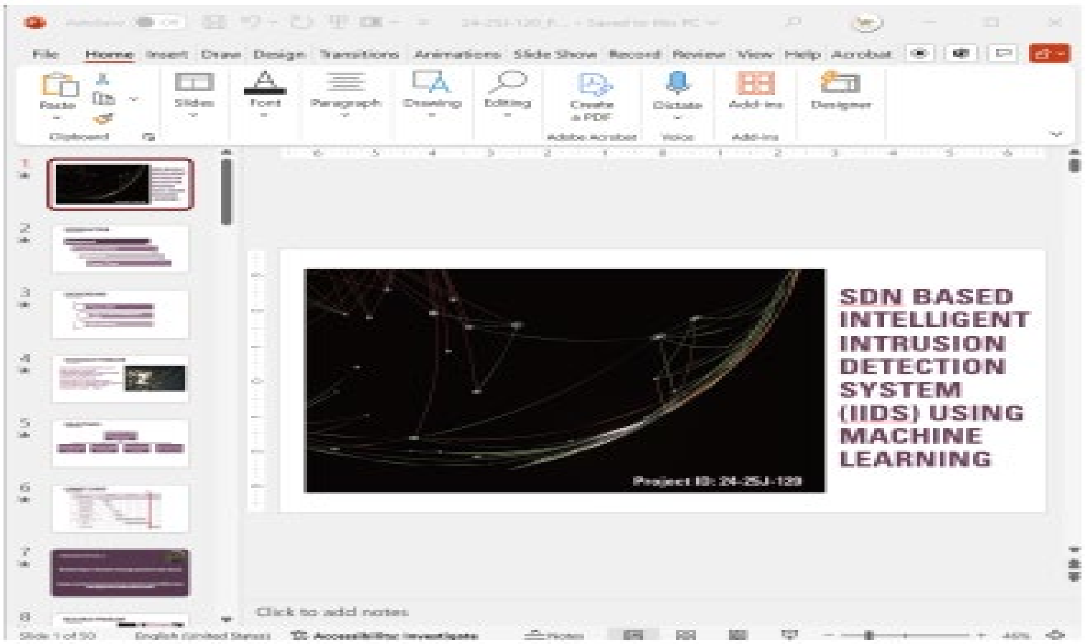

**Completed Task and Conversation Highlights**

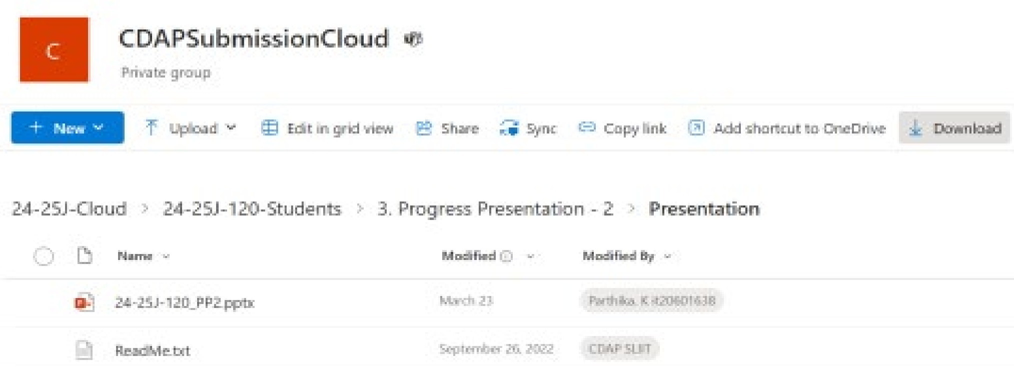
• Prepare for Progress Presentation 2 (PP2).

• Creating the presentation.

• Finalizing the Projects.

• Communication with the supervisor after finalizing the project



## Completed Task and Conversation Highlights

- Started writing the research paper.
- Exploring the IEEE standards and word tools.
- Communicating with supervisor and getting the supervisor feedback.

# SDN-based Intelligent Intrusion Detection System (IIDS) using Machine Learning


Parthika.K
*Faculty of Computing*
*Cyber Security Specilization*
*Sri Lanka Institute of Information Technology*
Malabe, Sri Lanka
kparthika@gmail.com

Satkurulingam.S
*Faculty of Computing*
*Cyber Security Specilization*
*Sri Lanka Institute of Information Technology*
Malabe, Sri Lanka
savithurisatkurulingam@gmail.com

Sriskandarajah J.P
*Faculty of Computing*
*Cyber Security Specilization*
*Sri Lanka Institute of Information Technology*
Malabe, Sri Lanka
srijoanna0@gmail.com

Dassanayake E.D
*Faculty of Computing*
*Cyber Security Specilization*
*Sri Lanka Institute of Information Technology*
Malabe, Sri Lanka
erangadassanayake15@gmail.com

Kanishka Prajeewa Yapa
*Department of Computer Systems Engineering*
*Sri Lanka Institute of Information Technology*
Malabe, Sri Lanka
kanishka.y@sliit.lk

Tharaniyawarna.K
*Department of Computer Systems Engineering*
*Sri Lanka Institute of Information Technology*
Malabe, Sri Lanka
tharaniyawarna.k@sliit.lk



***Abstract*— The increasing network complexity needs Software-Defined Networking (SDN) as a key solution to establish effective management systems through dynamic control mechanisms. SDN network infrastructure encounters four main security threats including Denial of Service (DoS), Flow Table Overflow, SQLite and Topology Poisoning attacks. This paper presents IIDS as an SDN-based intelligent intrusion detection system that operates with machine learning schemes to identify threats during real-time interactions. A dynamic system links the SDN controller with machine learning models for network traffic analysis to detect anomalies. The testing of the proposed method generates results for accuracy while also measuring precision and recall along with F1-score values. Starting from optimized attack detection the implemented technology is confirmed as an effective security framework for SDN networks because of its precise outcome.**



**Keywords—cyber security, software defined networking, intrusion detection system, machine learning**


### I. Introduction

By implementing Software-Defined Networking (SDN) operators can execute dynamic management operations combined with automated system management tasks for their network infrastructure [1]. Multiple security threats can occur through SDN's central management architecture because it exposes itself to various cyber-attacks. Network resilience becomes unsustainable due to difficulties with implementing

obtain live policy controls that improve network protection. Studies present evidence that SDN Technology integration with machine learning achieves successful threat detection solutions by using intelligent intrusion systems which improve security protection for network infrastructure.

### II. Literature Review

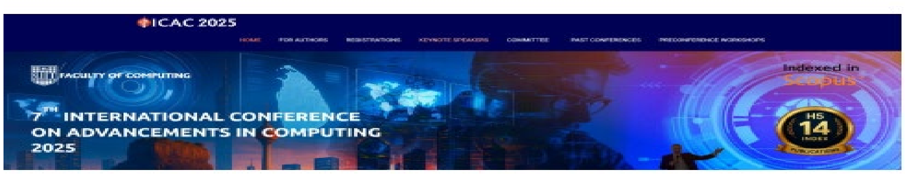
Studies performed by technologists demonstrate IDS technology as a solution to enhance SDN security while dynamic network administration needs advanced threat management solutions [3]. Traditional IDS operations heavily depend on signature detection thus their ability to detect modern cyber threats remains low [4]. Machine learning within IDS technology speeds up performance and adjusts to new threats because it detects unknown attacks by analyzing patterns and statistical deviations [5]. The application of machine learning in IDS achieves superior performance speed as well as adaptability through its ability to detect unknown threats through anomaly detection and pattern recognition capabilities. Numerous researchers have focused on using multivariate statistical analysis to detect SDN attacks, as this approach enhances network security detection capabilities. By analyzing multiple variables simultaneously, this technique improves anomaly detection, identifying potential security threats more accurately. Researchers have explored various statistical models to enhance intrusion detection systems, making them more effective in mitigating security risks within SDN environments [6]. The evaluation of Flow Table

**Looking to Publish Your Research and Meet RP Deadlines?**

**ICAC 2025 is the Opportunity!**

If you're searching for a **reputed and impactful venue** to publish your research paper and meet your Research Project (RP) deadlines, **ICAC 2025** is ready to support you.

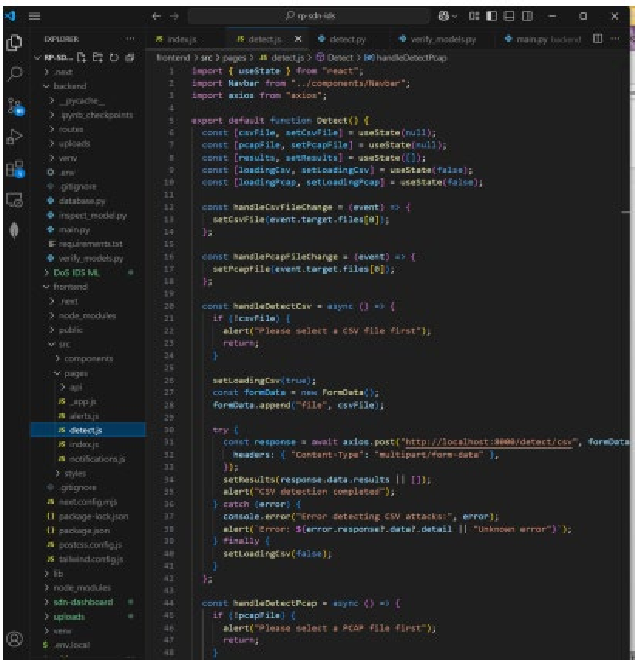
**Date:** Wednesday, **21st May 2025**
**Time:** 7.30 PM
**Speaker:** *Dr. Prasanna Sumathipala*
**Zoom Link:**

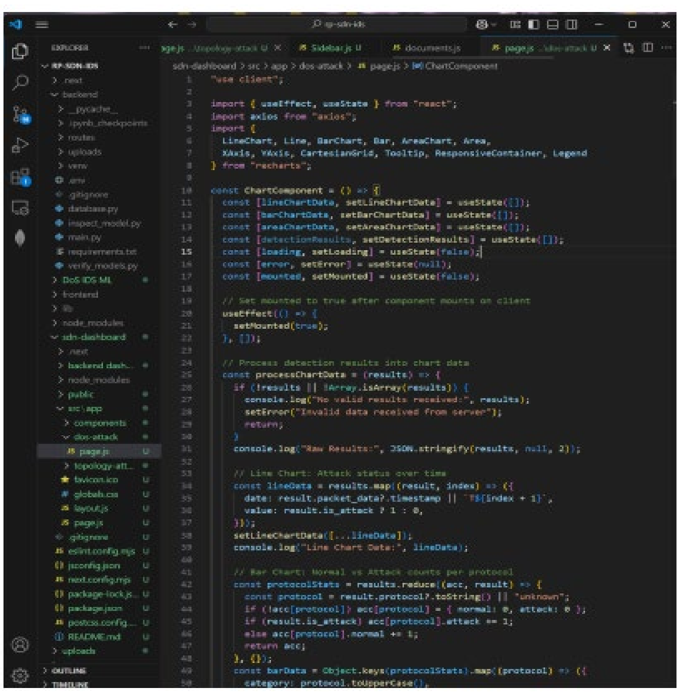




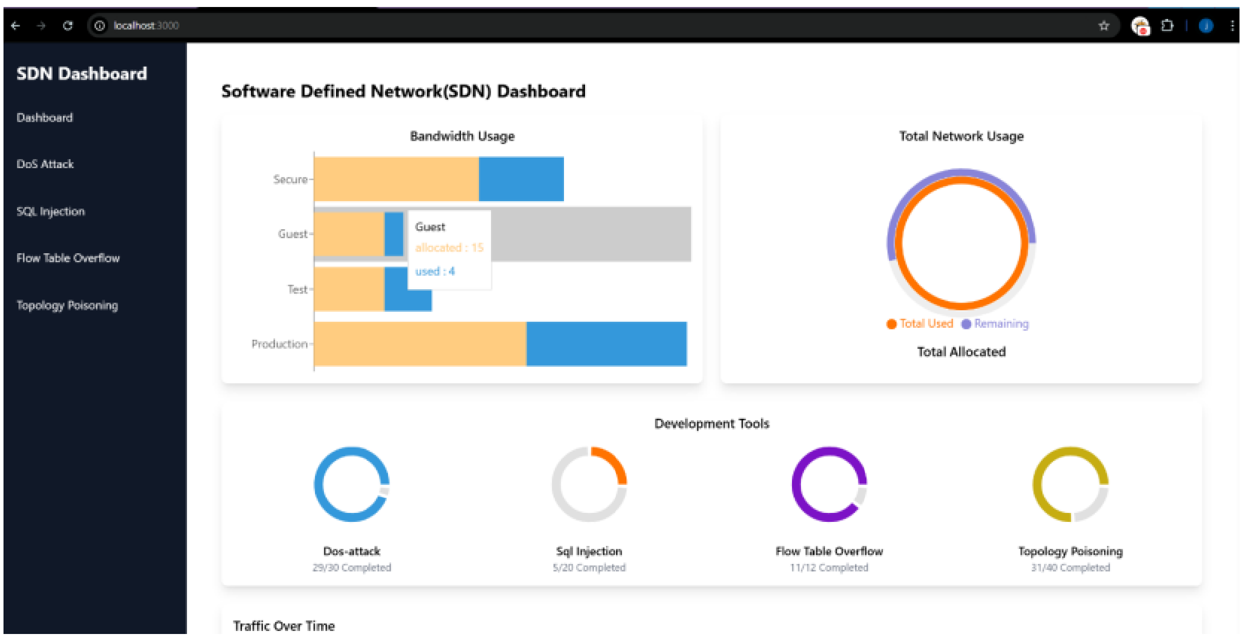
### Completed Tasks and Conversation Highlights

- Creating the front-end of the application.
- Integration of all the components.
- Discussing the supervisor’s suggestions






localhost:3001
SDN IDS
HOME
DETECT
NOTIFICATIONS
PAST
WELCOMETO SDN IDS


localhost:3000
SDN Dashboard
Dashboard
DoS Attack
SQL Injection
Flow Table Overflow
Topology Poisoning
Software Defined Network(SDN) Dashboard
Bandwidth Usage
Secure
Guest
Test
Production
Guest
allocated : 15
used : 4
Total Network Usage
Total Used
Remaining
Total Allocated
Development Tools
Dos-attack
29/30 Completed
Sql Injection
5/20 Completed
Flow Table Overflow
11/12 Completed
Topology Poisoning
31/40 Completed
Traffic Over Time

## Completed Tasks and Conversation Highlights

- Complete Individual Thesis Reports.
- Creation Group Thesis Reports

**SDN BASED INTRUSION DETECTION SYSTEM USING MACHINE LEARNING FOR TOPOLOGY POISONING ATTACK: A CASE STUDY**

**Satkurulingam.S**

**IT21282072**

BSc (Hons) degree in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2024

**DECLARATION**

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

| Name | Student ID | Signature |
|---|---|---|
| Satkurulingam. S | IT 21282072 | |

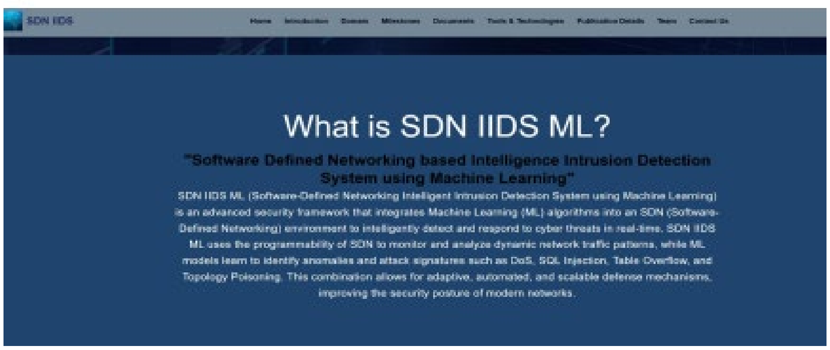
Signature of the Supervisor (Mr.Kanishka Yapa ) ..................................... Date ..............................

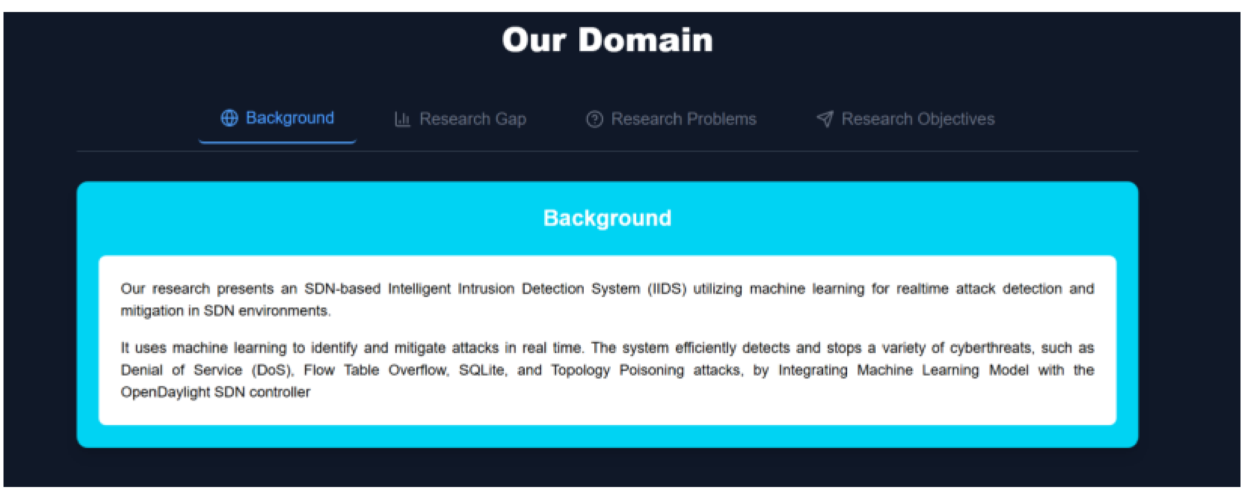
24-25J-Cloud > 24-25J-120-Students > 5. Final Report & Presentation > **Final Reports**

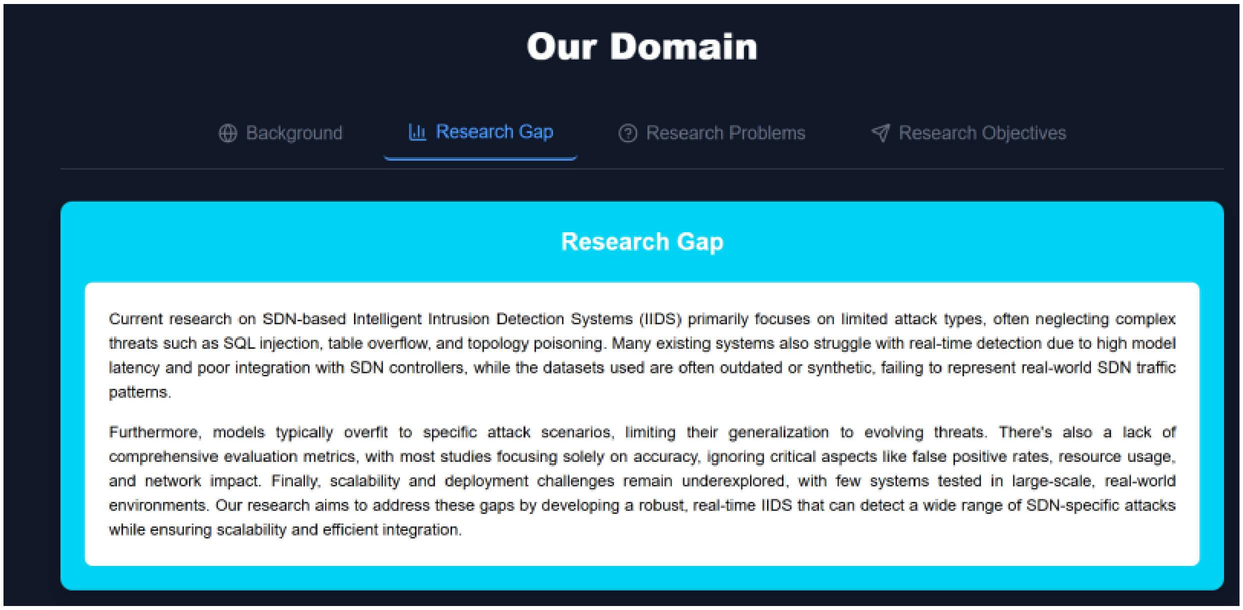
| Name | Modified | Modified By |
|---|---|---|
| TurnitIn reports | July 26, 2024 | CDAP SLIIT |
| IT20601638_Parthika.K_FinalReport.pdf | April 11 | Parthika. K it20601638 |
| IT21192982_Dassanayake E.D_Final Report.... | April 11 | Dassanayake E. D. it21192982 |
| IT21261978_Sriskandarajah J.P_Final_Repor... | April 12 | Sriskandarajah J.P it21261978 |
| IT21282072_Satkurulingam.S_FinalReport.p... | April 12 | Satkurulingam S it21282072 |
| ReadMe.txt | September 26, 2022 | CDAP SLIIT |
| RP_24-25J_120 -Final report.pdf | April 13 | Satkurulingam S it21282072 |

### Completed Tasks and Conversation Highlights

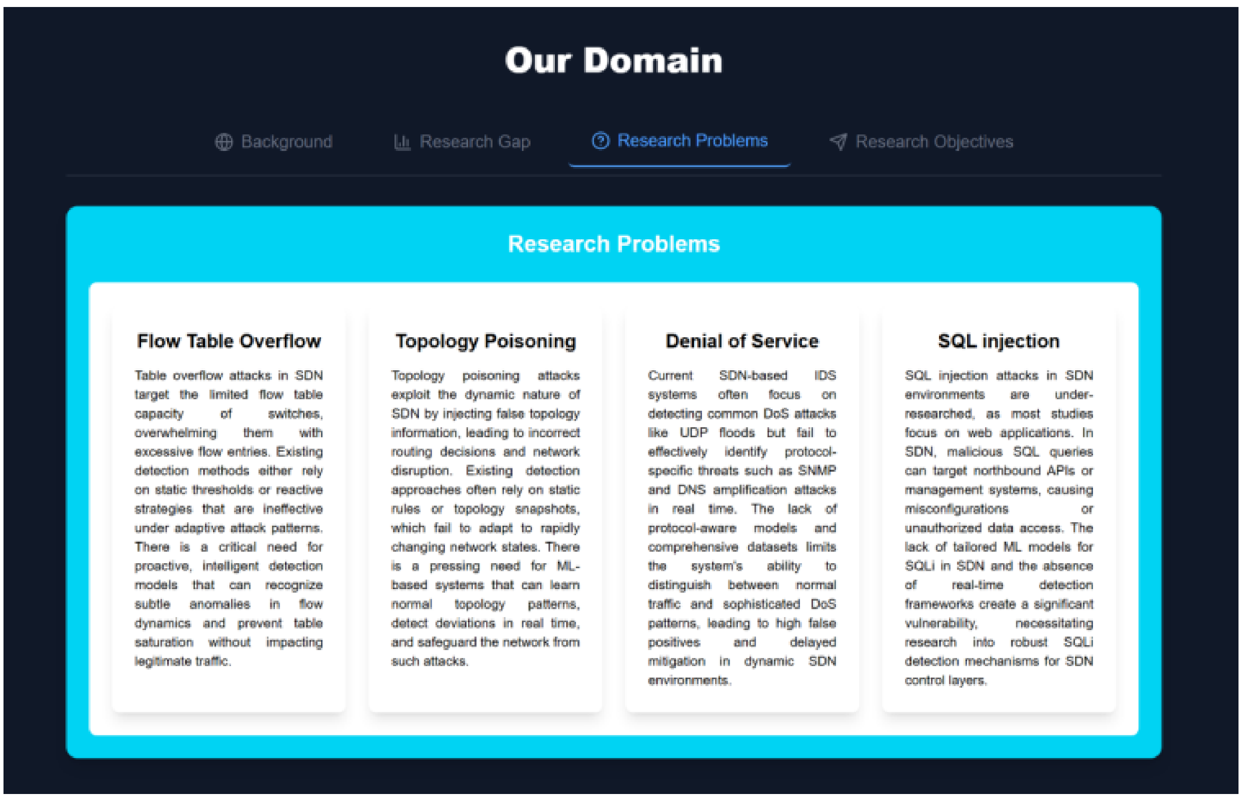
• Create a website for the solution

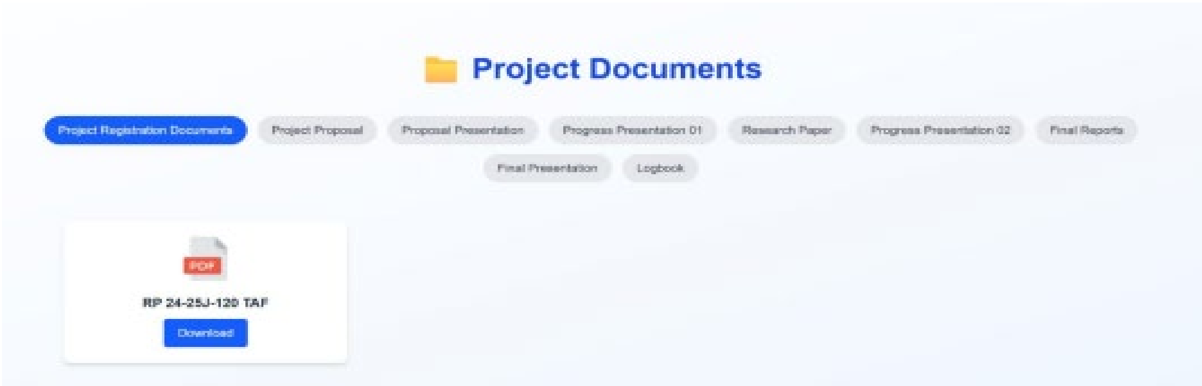



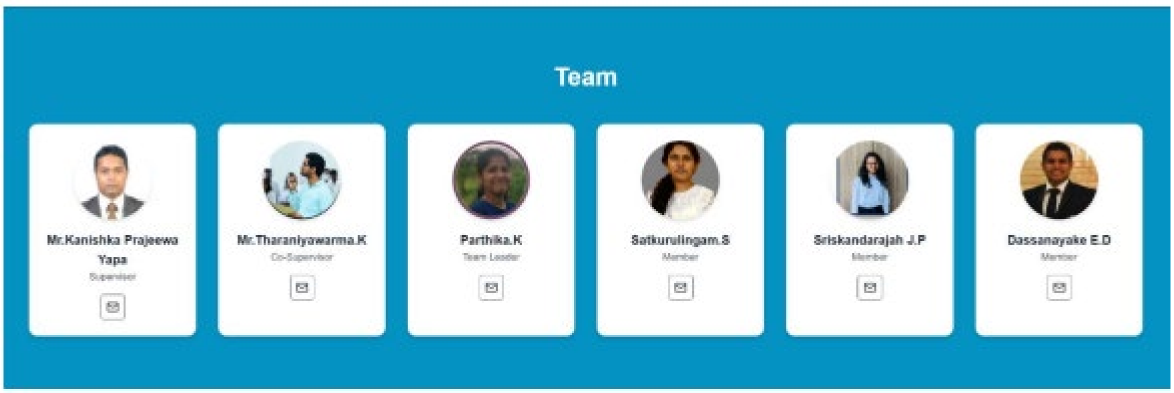


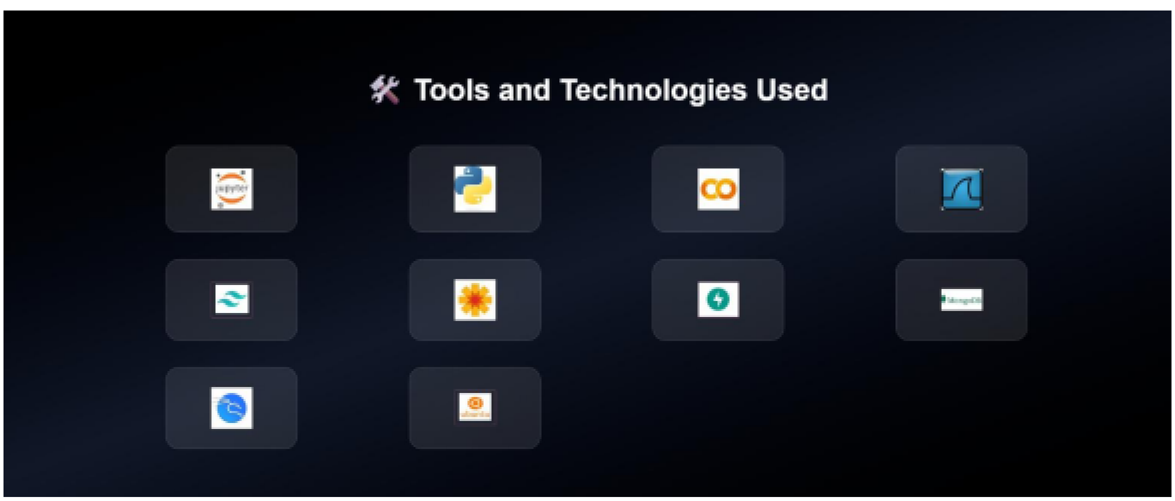
Our Domain
Background
Research Gap
Research Problems
Research Objectives
Research Gap
Current research on SDN-based Intelligent Intrusion Detection Systems (IIDS) primarily focuses on limited attack types, often neglecting complex threats such as SQL injection, table overflow, and topology poisoning. Many existing systems also struggle with real-time detection due to high model latency and poor integration with SDN controllers, while the datasets used are often outdated or synthetic, failing to represent real-world SDN traffic patterns.
Furthermore, models typically overfit to specific attack scenarios, limiting their generalization to evolving threats. There's also a lack of comprehensive evaluation metrics, with most studies focusing solely on accuracy, ignoring critical aspects like false positive rates, resource usage, and network impact. Finally, scalability and deployment challenges remain underexplored, with few systems tested in large-scale, real-world environments. Our research aims to address these gaps by developing a robust, real-time IIDS that can detect a wide range of SDN-specific attacks while ensuring scalability and efficient integration.

Our Domain
Background
Research Gap
Research Problems
Research Objectives
Research Problems
Flow Table Overflow
Table overflow attacks in SDN target the limited flow table capacity of switches, overwhelming them with excessive flow entries. Existing detection methods either rely on static thresholds or reactive strategies that are ineffective under adaptive attack patterns. There is a critical need for proactive, intelligent detection models that can recognize subtle anomalies in flow dynamics and prevent table saturation without impacting legitimate traffic.
Topology Poisoning
Topology poisoning attacks exploit the dynamic nature of SDN by injecting false topology information, leading to incorrect routing decisions and network disruption. Existing detection approaches often rely on static rules or topology snapshots, which fail to adapt to rapidly changing network states. There is a pressing need for ML-based systems that can learn normal topology patterns, detect deviations in real time, and safeguard the network from such attacks.
Denial of Service
Current SDN-based IDS systems often focus on detecting common DoS attacks like UDP floods but fail to effectively identify protocol-specific threats such as SNMP and DNS amplification attacks in real time. The lack of protocol-aware models and comprehensive datasets limits the system's ability to distinguish between normal traffic and sophisticated DoS patterns, leading to high false positives and delayed mitigation in dynamic SDN environments.
SQL injection
SQL injection attacks in SDN environments are under-researched, as most studies focus on web applications. In SDN, malicious SQL queries can target northbound APIs or management systems, causing misconfigurations or unauthorized data access. The lack of tailored ML models for SQLi in SDN and the absence of real-time detection frameworks create a significant vulnerability, necessitating research into robust SQLi detection mechanisms for SDN control layers.

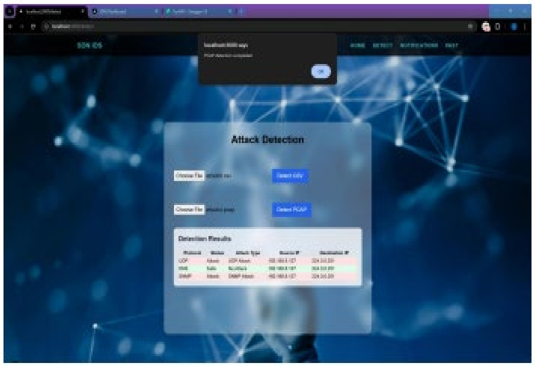

Project Documents
Project Registration Documents
Project Proposal
Proposal Presentation
Progress Presentation 01
Research Paper
Progress Presentation 02
Final Reports
Final Presentation
Logbook
PDF
RP 24-25J-120 TAF
Download


Team
Mr.Kanishka Prajeewa Yapa
Supervisor
Mr.Tharaniyawarma.K
Co-Supervisor
Parthika.K
Team Leader
Satkurulingam.S
Member
Sriskandarajah J.P
Member
Dassanayake E.D
Member

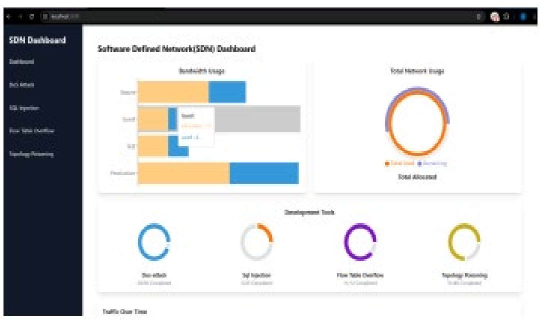

Tools and Technologies Used

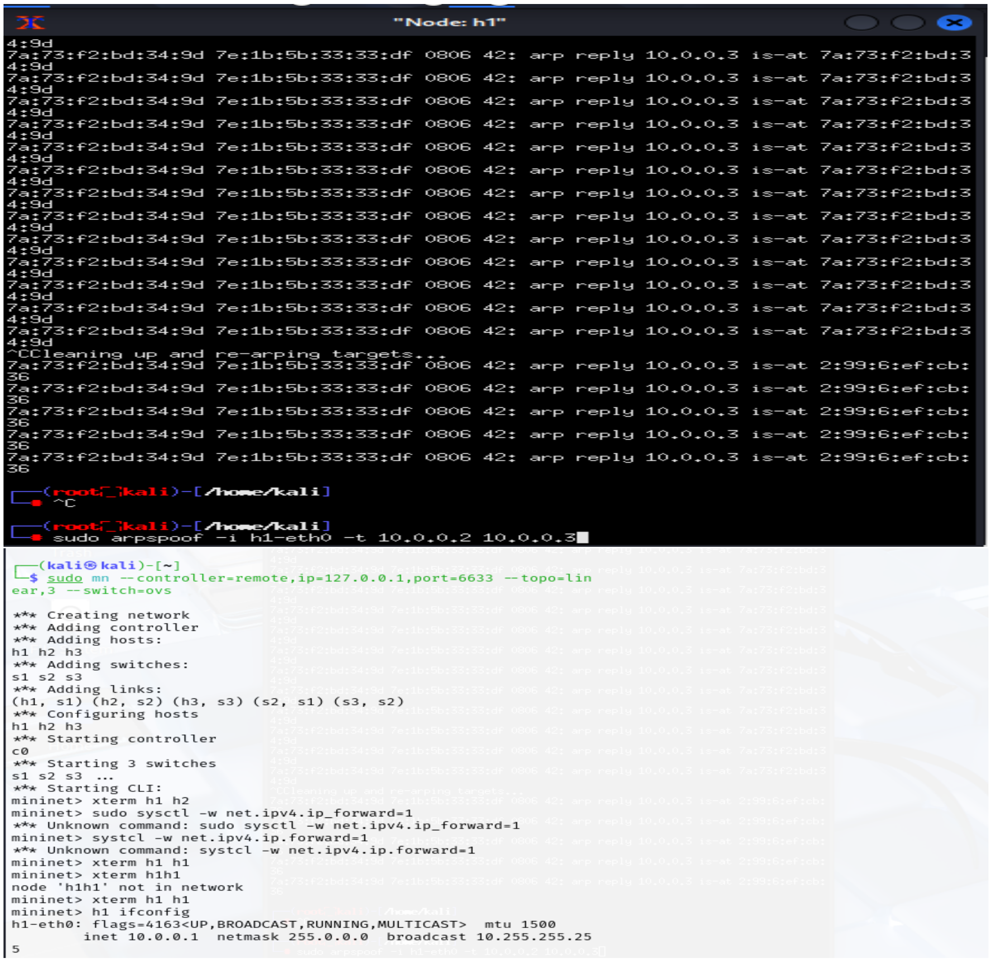
## Completed Tasks and Conversation Highlights

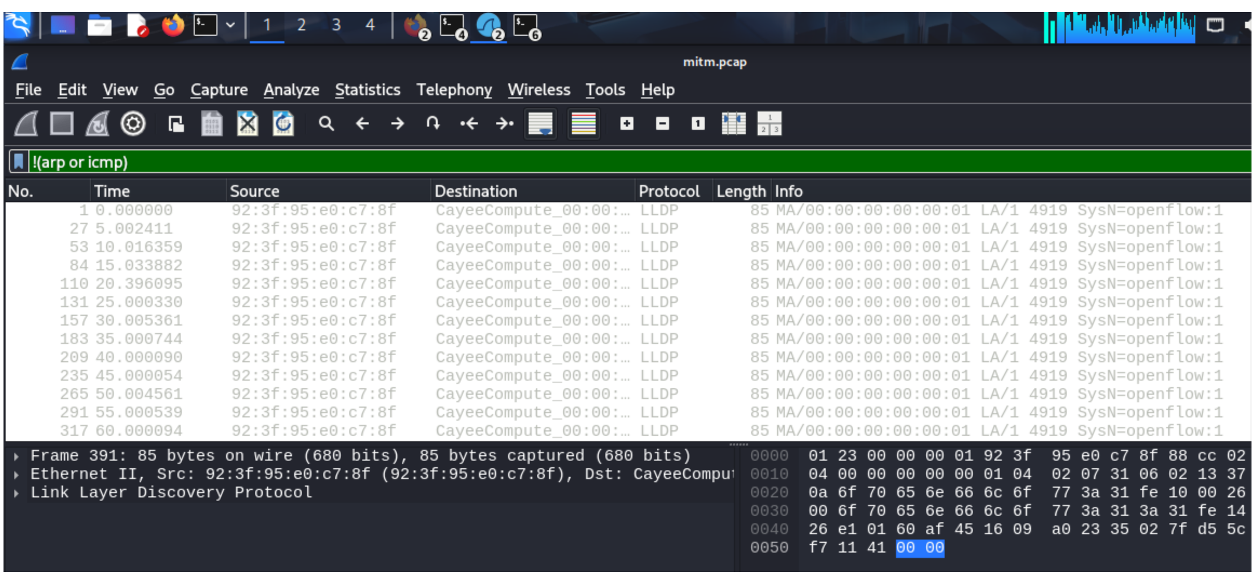
• Final Research Project Product





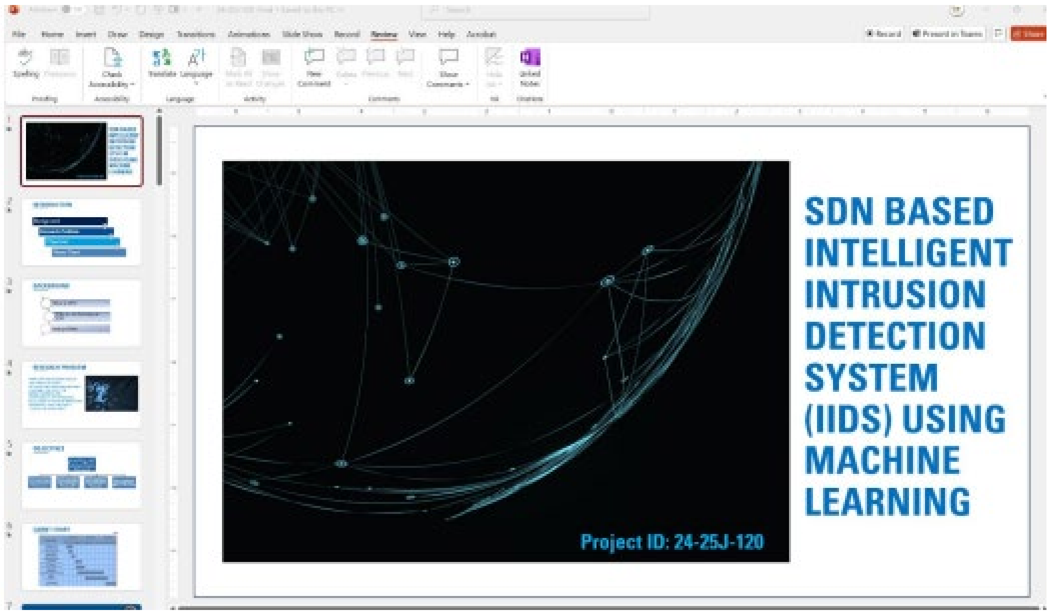



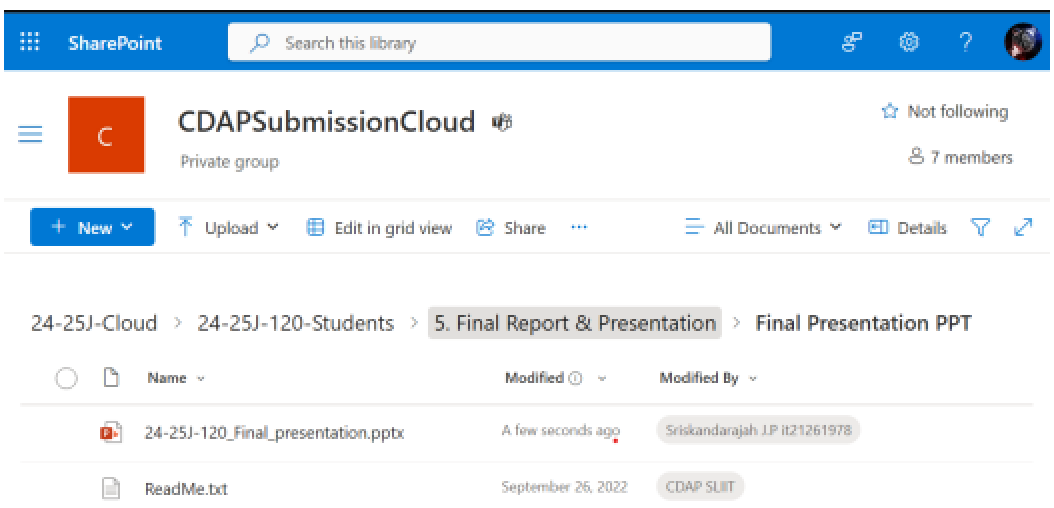

"Node: h1"
4:9d
7a:73:f2:bd:34:9d 7e:1b:5b:33:33:df 0806 42: arp reply 10.0.0.3 is-at 7a:73:f2:bd:3
^CCleaning up and re-arping targets...
7a:73:f2:bd:34:9d 7e:1b:5b:33:33:df 0806 42: arp reply 10.0.0.3 is-at 2:99:6:ef:cb:
36
(root㉿kali)-[/home/kali]
^C
(root㉿kali)-[/home/kali]
sudo arpspoof -i h1-eth0 -t 10.0.0.2 10.0.0.3
(kali㉿kali)-[~]
$ sudo mn --controller=remote,ip=127.0.0.1,port=6633 --topo=linear,3 --switch=ovs
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s2, s1) (s3, s2)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet> sudo sysctl -w net.ipv4.ip_forward=1
*** Unknown command: sudo sysctl -w net.ipv4.ip_forward=1
mininet> systcl -w net.ipv4.ip.forward=1
*** Unknown command: systcl -w net.ipv4.ip.forward=1
mininet> xterm h1 h1
mininet> xterm h1h1
node 'h1h1' not in network
mininet> xterm h1 h1
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255


mitm.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
!(arp or icmp)
No. Time Source Destination Protocol Length Info
1 0.000000 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
27 5.002411 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
53 10.016359 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
84 15.033882 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
110 20.396095 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
131 25.000330 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
157 30.005361 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
183 35.000744 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
209 40.000090 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
235 45.000054 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
265 50.004561 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
291 55.000539 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
317 60.000094 92:3f:95:e0:c7:8f CayeeCompute_00:00:… LLDP 85 MA/00:00:00:00:00:01 LA/1 4919 SysN=openflow:1
Frame 391: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)
Ethernet II, Src: 92:3f:95:e0:c7:8f (92:3f:95:e0:c7:8f), Dst: CayeeCompu
Link Layer Discovery Protocol
0000 01 23 00 00 00 01 92 3f 95 e0 c7 8f 88 cc 02
0010 04 00 00 00 00 00 01 04 02 07 31 06 02 13 37
0020 0a 6f 70 65 6e 66 6c 6f 77 3a 31 fe 10 00 26
0030 00 6f 70 65 6e 66 6c 6f 77 3a 31 3a 31 fe 14
0040 26 e1 01 60 af 45 16 09 a0 23 35 02 7f d5 5c
0050 f7 11 41 00 00

### Completed Task and Conversation Highlights

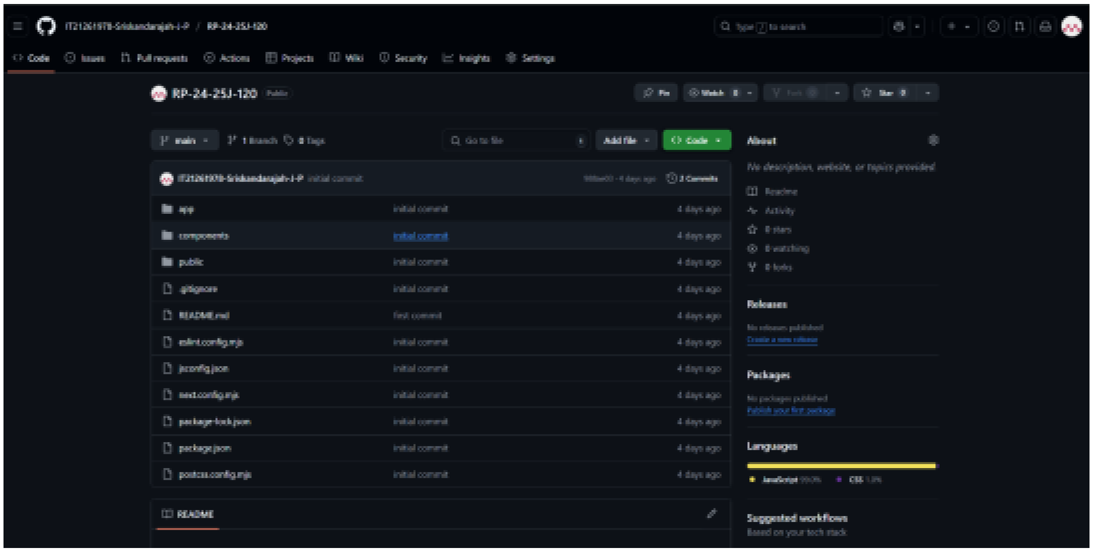
- Prepare for Final Presentation
- Creating the presentation.



## Completed Task and Conversation Highlights

• Commit and push the website codes in GitHub before deploying




## Completed Task and Conversation Highlights

• Deploy the website using vercel