AUTOMATED REAL-TIME CYBER THREAT DETECTION AND MITIGATION WITH MACHINE LEARNING

***Project Reference No.**: 48S_BE_2062*

College : Vivekananda Institute of Technology, Bengaluru
Branch : Department of Information Science and Engineering
Guide(s) : Dr. P Vanajakshi
Student(s) : Mr. Mahesh A V
Mr. Lakshmi Prasad V
Mr. Abhishek G
Ms. Anusha P B

Keywords:

Cybersecurity, Real-Time Threat Detection, Machine Learning, Anomaly Detection, Zero-Day Exploit.

Introduction:

The ongoing shift to a digital world has thrown home networks, small businesses, and companies into a pool of threats that are versatile enough to contain any transfer of cyberattacks. The thrusting rise of smart devices, cloud services, and online platforms is causing manifold expansions in the attack surfaces for malicious actors. Unlike large organisations, home users and small businesses are not equipped with IT teams and have no enterprise-grade security infrastructure, making it very easy for cyber criminals to exploit them.

As a furtherance in the title of this project: Automated Real-Time Cyber Threat Detection and Mitigation with Machine Learning shall aim at the provision of the proper security apparatus essential to a home network and small business environment. The ultimate goal is to build a cheap intelligent, fully automated solution that will be able to detect and mitigate a mix of common and advanced cyber-attacks in real time while keeping minimal human intervention as its requirement.

Many attacks hit our system, and these include some Distributed Denial of Service (DDoS) attacks, SQL Injection, Brute Force, and Zero-Day Exploits which can be regarded as attacks capable of crippling business operations and possible data leakage. In these scenarios, classification mechanisms based on a combination of supervised and unsupervised Machine Learning algorithms help in identifying known attack patterns for immediate response, as well as detecting hitherto unseen anomalies.

The project considers well-known datasets in cybersecurity for model training and validation, namely CICIDS2017, UNSW-NB15, and KDD99, assuring that real-world attacks can be recognized by the models with high confidence. The tools used in the project include Wireshark, PyShark, Tshark, and Scapy for packet-level traffic analysis and feature extraction.

The use of cheap hardware such as Raspberry Pi 4 Model B for running the trained models assures affordability and practicability to small-scale users by facilitating in-situ real-time monitoring and protection at the network perimeter. In the event of identifying a suspected attack, the system then employs Nmap to identify crucial details about the attacker including operating system, MAC address, public IP,

Objectives:

This project is to design and implement an intelligent, automated, and real-time cyber threat detection and mitigation system tailored specifically for home networks and small businesses. This project aims to reduce the dependency on manual network monitoring and costly enterprise-grade security solutions by integrating Machine Learning, network analysis, and automated response mechanisms into a single streamlined solution.

The key objectives of the project are as follows:

- To develop an automated system capable of detecting and classifying cyber-attacks such as Distributed Denial of Service (DDoS), SQL Injection, Brute Force attacks, and Zero-Day Exploits using both supervised and unsupervised Machine Learning models.
- To implement a hybrid detection pipeline that can handle both known attack patterns and unknown anomalies, ensuring the system’s adaptability to evolving cyber threats.
- To train and validate detection models using established and diverse datasets such as CICIDS2017, UNSW-NB15, and KDD99, ensuring robustness and reliability in real-world environments.
- To deploy the detection system on a Raspberry Pi 4 Model B, demonstrating a cost-effective and scalable solution suitable for homes and small businesses.

- To automate the threat response process by integrating tools such as Nmap for device identification and iptables / nftables for active blocking of malicious hosts and IP addresses.
- To enhance incident visibility by integrating a Telegram Bot that sends real-time alerts, ensuring that administrators and users are promptly informed of threats and mitigation actions.
- To design a system that requires minimal human supervision, enabling proactive cybersecurity defence even for users with limited technical knowledge.

Methodology:

This project involves a systematic approach to designing, training, deploying, and evaluating a real-time cyber threat detection and mitigation system using both supervised and unsupervised machine learning algorithms. The system is built with the intent to secure home networks and small businesses by automating the detection and defence against cyber threats.

Materials Used

- Datasets:
    - CICIDS2017
    - UNSW-NB15
    - KDD99
- Software Tools:
    - Wireshark, PyShark, Tshark, and Scapy (for packet capture and feature extraction)
    - Nmap (for network reconnaissance and attacker fingerprinting)
    - iptables / nftables (for traffic filtering and mitigation)
    - Python (for model development and automation)
    - Telegram Bot API (for real-time alerting)
- Hardware:
    - Raspberry Pi 4 Model B (8GB RAM) for deployment.

Methods and Workflow

1. Data Collection and Preprocessing:- The project starts with using publicly available cybersecurity datasets (CICIDS2017, UNSW-NB15, and KDD99) that

cover a wide range of modern and legacy cyber-attacks. Data cleaning, feature extraction, and transformation are performed to ensure the datasets are ready for training machine learning models.

2. Feature Extraction from Network Traffic:- Live network traffic is captured using Wireshark and its Python APIs — PyShark, Tshark, and Scapy. Features such as packet length, flow duration, protocol type, source/destination IPs, and flag counts are extracted for real-time analysis.
3. Model Training and Evaluation:- Multiple machine learning algorithms are applied to train the system for attack detection:
    - DDoS: K-Means Clustering (unsupervised) and k-Nearest Neighbors (supervised).
    - SQL Injection: Random Forest Classifier.
    - Brute Force Attack: Random Forest Classifier.
    - Zero-Day Exploit: Hybrid anomaly detection pipeline combining K-Means, KNN, Isolation Forest (unsupervised) and Random Forest, XGBoost (supervised).
    - The models are evaluated using accuracy, precision, recall, and F1-score to ensure reliable detection performance.
4. Deployment on Raspberry Pi:- The trained models are deployed on Raspberry Pi 4 Model B, transforming it into a compact, dedicated network security node capable of real-time packet analysis and anomaly detection.
5. Threat Detection and Mitigation:- Once deployed, the system monitors live network traffic. When suspicious activity is detected:
    - The system uses Nmap to perform reconnaissance, identifying the attacker's operating system, MAC address, public IP, geolocation, and ISP.
    - Detected malicious IP addresses are immediately blocked using iptables or nftables, preventing further communication from the source and neutralizing the threat at the network firewall level.
    - To ensure uninterrupted and parallel execution of traffic monitoring, detection, mitigation scripts, and logging, the project utilizes tmux (Terminal Multiplexer). This enables the Raspberry Pi to maintain

persistent sessions, automate multiple tasks, and resume processes even after disconnection or reboot.

6. Alerting Mechanism:- The system integrates a Telegram Bot to send real-time notifications containing attack details, IP information, and action logs to the network administrator or homeowner, enabling remote awareness and control.

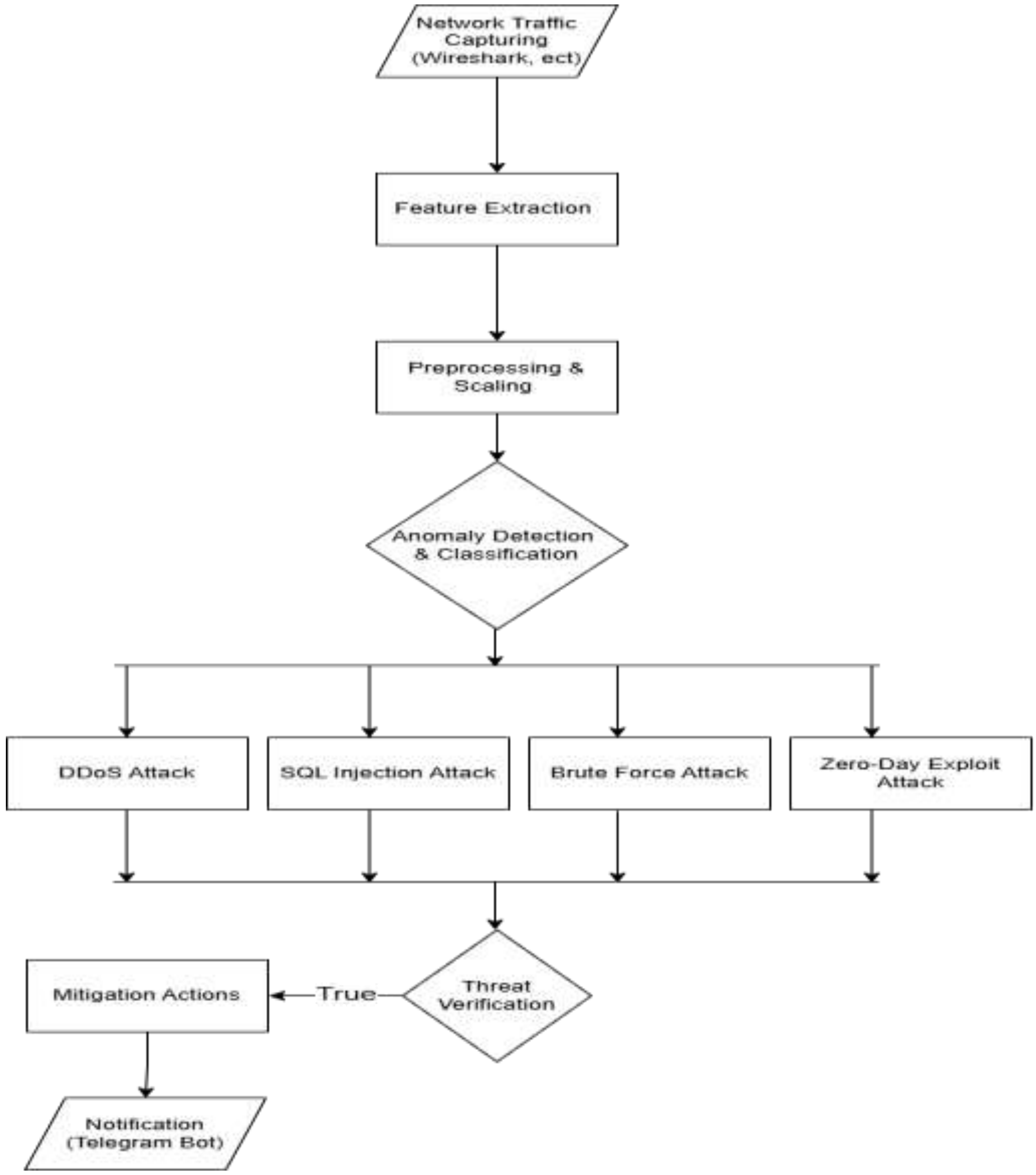


Figure 1: Flow Diagram

Result and Conclusion:

The implemented system successfully detected and classified multiple types of cyberattacks including Denial of Service (DoS/DDoS), SQL Injection, Brute Force

Attacks, and Zero-Day Exploits using a combination of supervised and unsupervised machine learning algorithms.

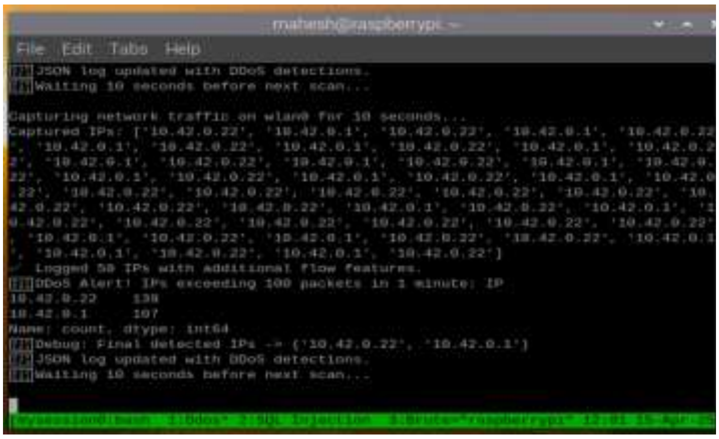
The use of datasets like CICIDS2017, UNSW-NB15, and KDD99 provided diverse and real-world network traffic scenarios that helped the system achieve high accuracy and low false positive rates during testing. The system was able to detect Zero-Day anomalies with the help of Isolation Forest, K-Means, KNN, and XGBoost models.

The deployment on Raspberry Pi 4 Model B demonstrated that the solution is not only efficient but also suitable for resource-constrained environments. Real-time packet capturing tools, such as Wireshark, Pyshark, Tshark, and Scapy, played a crucial role in feeding live traffic data to the models.

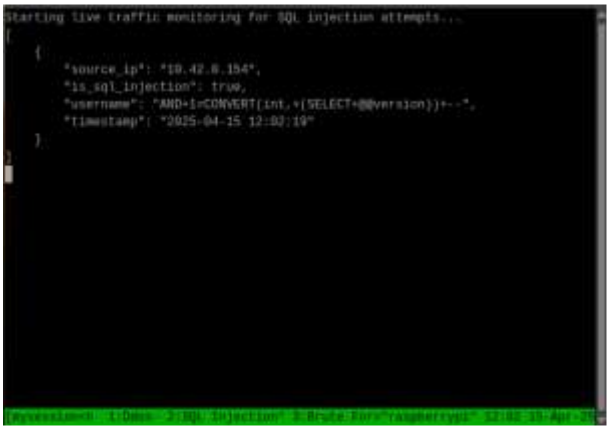
The system also successfully mitigated attacks by using Nmap for device discovery and iptables/nftables for blocking malicious IP addresses. The notification system, integrated with a Telegram bot, ensured immediate alerts were sent to the user upon detection of threats.

The project proved that a lightweight, automated, and intelligent cyber defense system can significantly improve response times to attacks while reducing human intervention. The combination of supervised and unsupervised learning allowed for the early detection of Zero-Day exploits, while the real-time blocking mechanisms helped prevent further damage from detected threats.

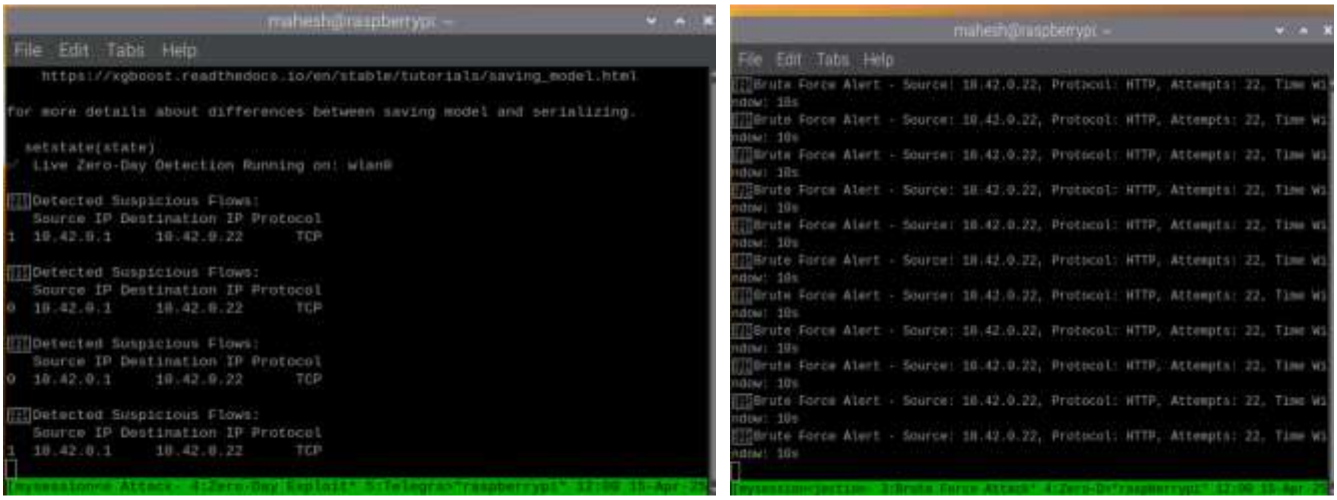
In conclusion, the project has demonstrated a practical and reliable approach for automated cyber threat detection and mitigation, highlighting its potential for further enhancements and real-world deployment in IoT, smart homes, and small business networks.



```
JSON log updated with DDoS detections.
Waiting 10 seconds before next scan...

Capturing network traffic on wlan0 for 10 seconds...
Captured IPs: ['10.42.0.22', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.22
', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.2
2', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.
22', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0
.22', '10.42.0.22', '10.42.0.22', '10.42.0.22', '10.42.0.22', '10.42.0.22', '10.
42.0.22', '10.42.0.22', '10.42.0.22', '10.42.0.1', '10.42.0.22', '10.42.0.1', '1
0.42.0.22', '10.42.0.22', '10.42.0.22', '10.42.0.22', '10.42.0.22', '10.42.0.22'
, '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.22', '10.42.0.22', '10.42.0.1
', '10.42.0.1', '10.42.0.22', '10.42.0.1', '10.42.0.22']
✓  Logged 50 IPs with additional flow features.
DDoS Alert! IPs exceeding 100 packets in 1 minute: IP
10.42.0.22     138
10.42.0.1      107
Name: count, dtype: int64
Debug: Final detected IPs -> ('10.42.0.22', '10.42.0.1')
JSON log updated with DDoS detections.
Waiting 10 seconds before next scan...

[mysession0:bash  1:Ddos* 2:SQL Injection  3:Brute-"raspberrypi" 12:01 15-Apr-25
```



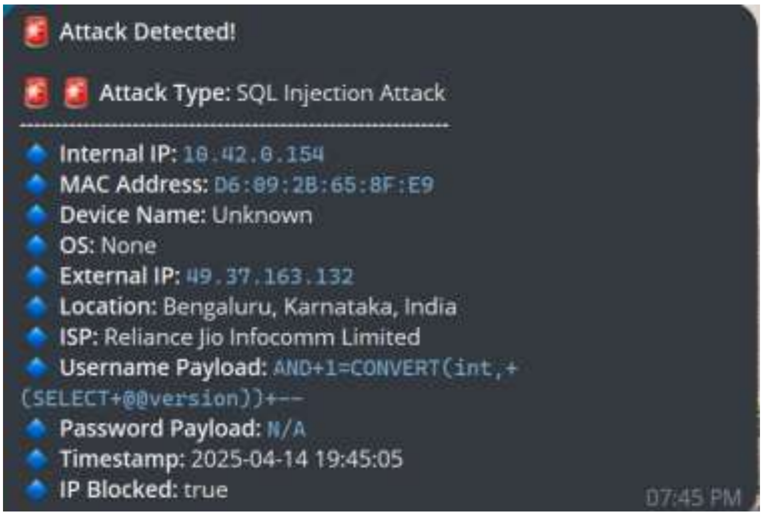
```
Starting live traffic monitoring for SQL injection attempts...
[
    {
        "source_ip": "10.42.0.154",
        "is_sql_injection": true,
        "username": "AND+1=CONVERT(int,+(SELECT+@@version))+--",
        "timestamp": "2025-04-15 12:02:19"
    }
]

[mysession0  1:Ddos  2:SQL Injection* 3:Brute For*"raspberrypi" 12:02 15-Apr-25
```



```
   https://xgboost.readthedocs.io/en/stable/tutorials/saving_model.html

for more details about differences between saving model and serializing.

  setstate(state)
✓ Live Zero-Day Detection Running on: wlan0

Detected Suspicious Flows:
   Source IP Destination IP Protocol
1  10.42.0.1     10.42.0.22       TCP

Detected Suspicious Flows:
   Source IP Destination IP Protocol
0  10.42.0.1     10.42.0.22       TCP

Detected Suspicious Flows:
   Source IP Destination IP Protocol
0  10.42.0.1     10.42.0.22       TCP

Detected Suspicious Flows:
   Source IP Destination IP Protocol
1  10.42.0.1     10.42.0.22       TCP

[mysession]0 Attack- 4:Zero-Day Exploit* 5:Telegra>"raspberrypi" 12:00 15-Apr-25
```

```
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s
Brute Force Alert - Source: 10.42.0.22, Protocol: HTTP, Attempts: 22, Time Wi
ndow: 10s

[mysession-jection 3:Brute Force Attack' 4:Zero-D>"raspberrypi" 12:00 15-Apr-25
```

Figure 2: Detection of Attacks



🚨 **Attack Detected!**

🚨 🚨 **Attack Type:** SQL Injection Attack

🔹 **Internal IP:** 10.42.0.154
🔹 **MAC Address:** D6:89:2B:65:8F:E9
🔹 **Device Name:** Unknown
🔹 **OS:** None
🔹 **External IP:** 49.37.163.132
🔹 **Location:** Bengaluru, Karnataka, India
🔹 **ISP:** Reliance Jio Infocomm Limited
🔹 **Username Payload:** AND+1=CONVERT(int,+(SELECT+@@version))+--
🔹 **Password Payload:** N/A
🔹 **Timestamp:** 2025-04-14 19:45:05
🔹 **IP Blocked:** true

07:45 PM

Figure 3: Telegram Output

Future Scope:

- Expansion of Threat Detection: The system can be extended to detect additional complex cyber threats such as ransomware, phishing attempts, and insider attacks, which require deeper behavioural analysis and advanced pattern recognition.
- Integration with Deep Learning Models: Future versions can implement deep learning techniques like LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Networks) to enhance accuracy, particularly for detecting sophisticated and evolving attack patterns.
- Adaptive and Intelligent Mitigation: Current mitigation uses predefined firewall rules; future work can focus on developing adaptive firewall systems that utilize

real-time machine learning feedback to automatically adjust blocking and allowing policies without human intervention

- Automated Self-Training Models: Future systems can implement self-learning algorithms that retrain themselves on new datasets and traffic patterns, enhancing their detection capabilities over time without manual intervention.
- Multi-Platform Notification System: In addition to Telegram alerts, future work may include integrating notifications via email, mobile apps, and desktop alerts, making real-time security updates more accessible to users and administrators.