IMG_256

**Detection and Response: Enhancing Intrusion Detection Systems**

by:

Group 5

**Joemar Lugtu - 301355179**

**Mark Labiano - 301346652**

**Syed - 301318212**

**Tarun Kumar Ravikumar – 301364055**

Project Supervisor:

**Dr. Sk Md Mizanur Rahman**

for:

**2024 SETAS Technology Fair**

**Centennial College**

**School of Engineering Technology and Applied Science**

on:

**April 10, 2023.**

# Project Description

**The Issue**

In today's digital age, cybersecurity threats continue to evolve, presenting significant challenges for organizations worldwide. Among these threats, brute force attacks stand out as a particularly insidious method employed by malicious actors to gain unauthorized access to systems and sensitive data. These attacks involve systematic trial-and-error attempts to guess passwords or encryption keys, exploiting vulnerabilities in authentication systems. The consequences of successful brute force attacks can be catastrophic, ranging from data breaches and financial losses to reputational damage and legal liabilities.

**Objective and Methods**

Our project's main objective is to enhance intrusion detection systems to effectively detect brute force attacks in real-time industrial environments. To achieve this, we adopted a systematic approach, combining traditional cybersecurity methodologies with cutting-edge machine-learning techniques.

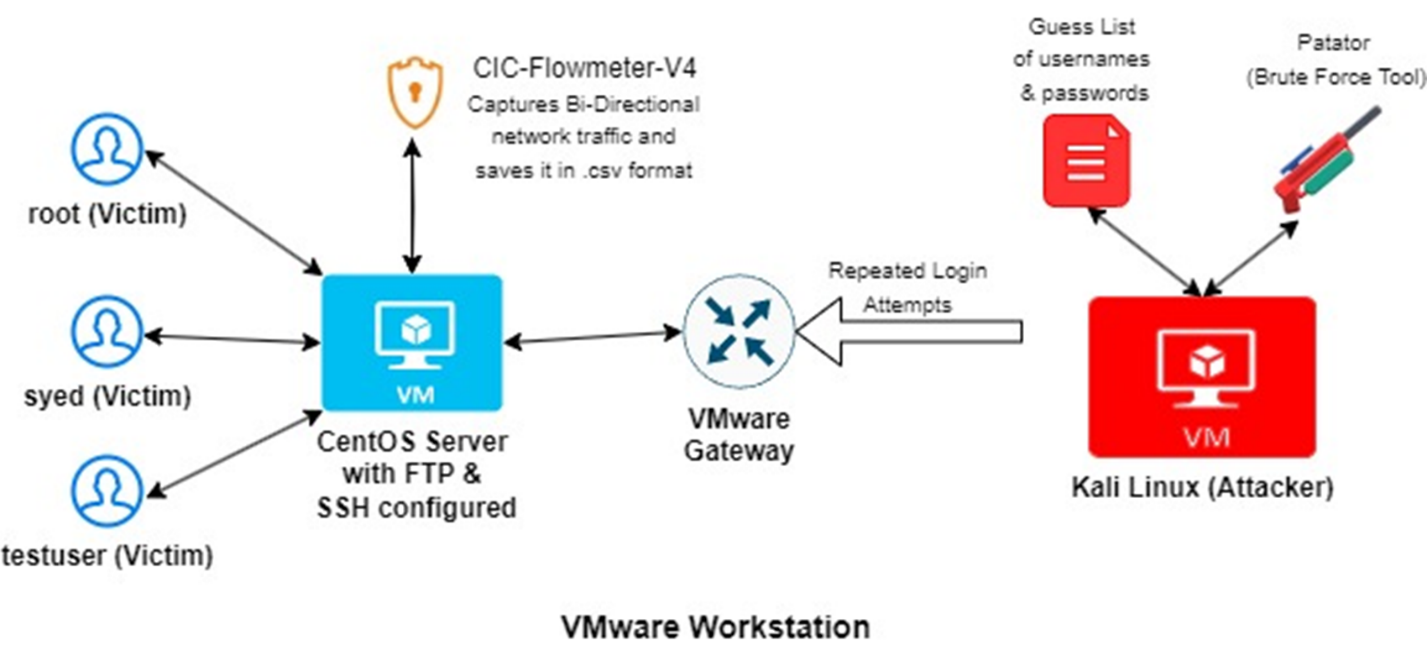


Figure 1 - Our Experimental Setup

In this setup, there are two virtual machines hosted on VMware Workstation. The CentOS machine acts as the victim machine with FTP and SSH services enabled. On the other hand, Kali Linux acts as the attacker machine, equipped with a guest list of commonly used usernames and passwords. The attacker machine needs to establish a connection with the victim machine to execute the brute force attack. The attack is facilitated using the built-in Kali tool, Patator, targeting the CentOS machine with the guest list. Additionally, the CentOS server hosts the CIC flow meter, capturing bidirectional network traffic information and storing it in a CSV format for further analysis. This setup simulates a controlled environment for testing and analyzing brute-force attack scenarios.

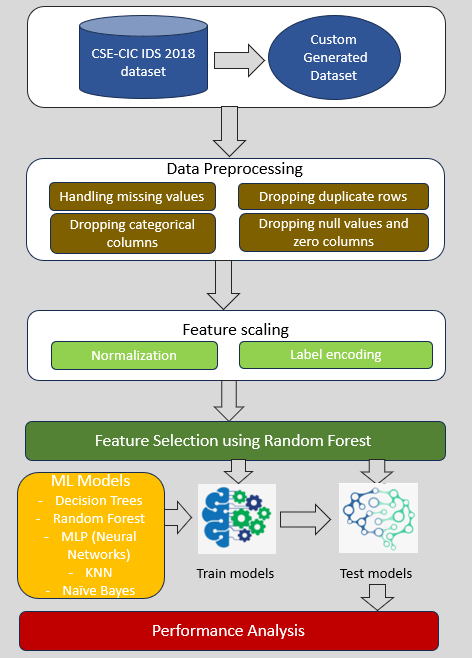


Figure 2 - Our Approach

Our methodology comprises several key steps:

* **Data Collection**: We collected network traffic data from a CIC flow meter hosted on a CentOS server. This data provided insights into the bidirectional flow of network traffic, enabling us to identify patterns indicative of brute-force attacks.
* **Data Preprocessing**: Before analysis, we cleaned and prepared the dataset to ensure data integrity and consistency. This involved removing irrelevant columns, handling missing values, and standardizing data formats.
* **Feature Scaling**: To enhance the effectiveness of our models, we employed feature scaling techniques such as MinMax scaling and encode categorical or textual output labels into numerical representations to ensure that all features contributed proportionally to the overall prediction.
* **Feature Selection**: With a multitude of features available, we prioritized the most influential ones using feature importance functions. This step allowed us to identify and focus on key features crucial for detecting brute force attacks.

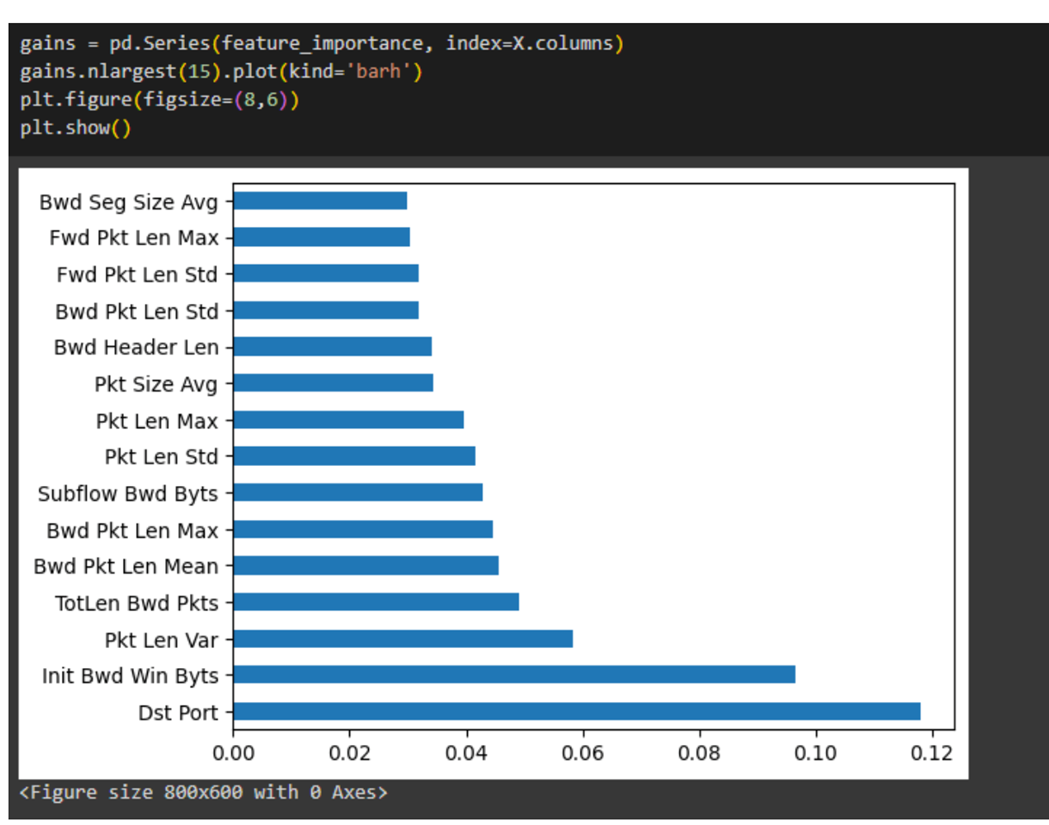


Figure 3 - 83 features from the CIC flow meter narrowed down to 15 key features

* **Model Training**: We trained multiple machine learning models, including Decision Tree, Random Forest, Neural Network, Naive Bayes, and KNN, to learn patterns indicative of malicious activities. Each model was meticulously fine-tuned to maximize prediction accuracy.
* **Performance Analysis**: Finally, we rigorously evaluated the performance of each model using metrics such as precision, recall, F1 score, accuracy, false positive rate and false negative rate. This comprehensive analysis allowed us to identify the most effective model for real-world deployment. Then, an analysis of the processing time for both training and prediction phases was done.

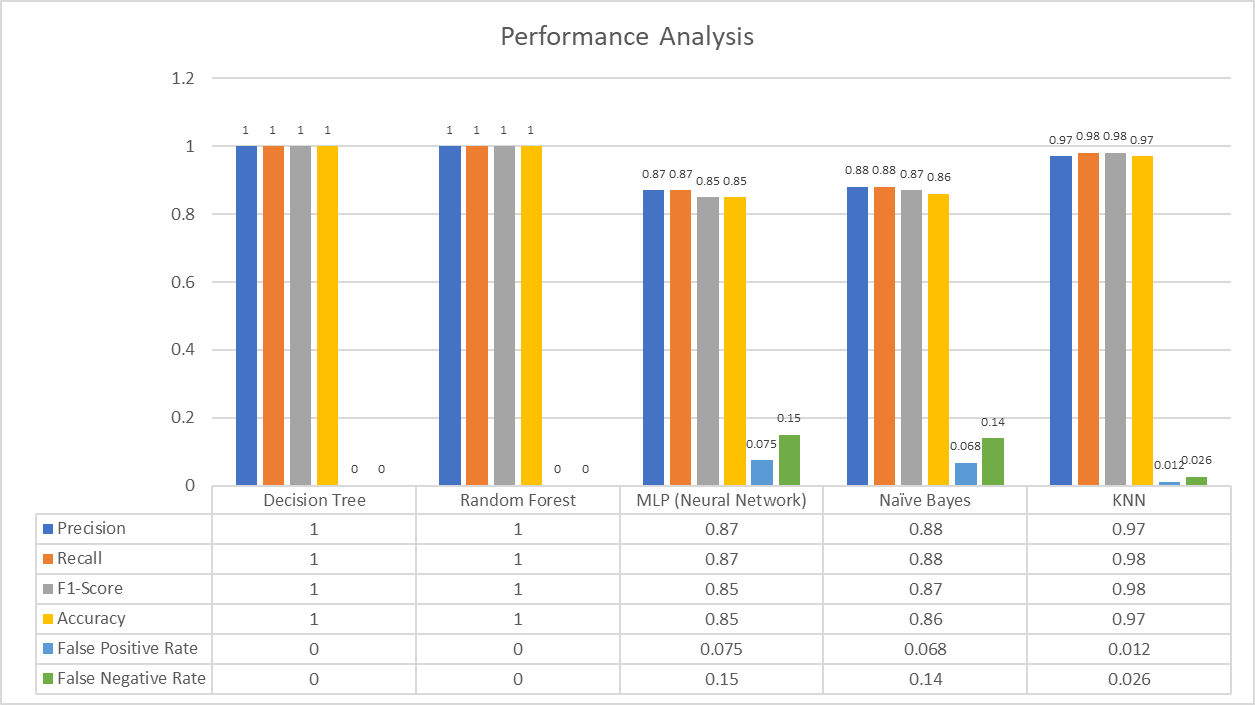


Figure 4 - Comparative Performance Metrics of Machine Learning Models

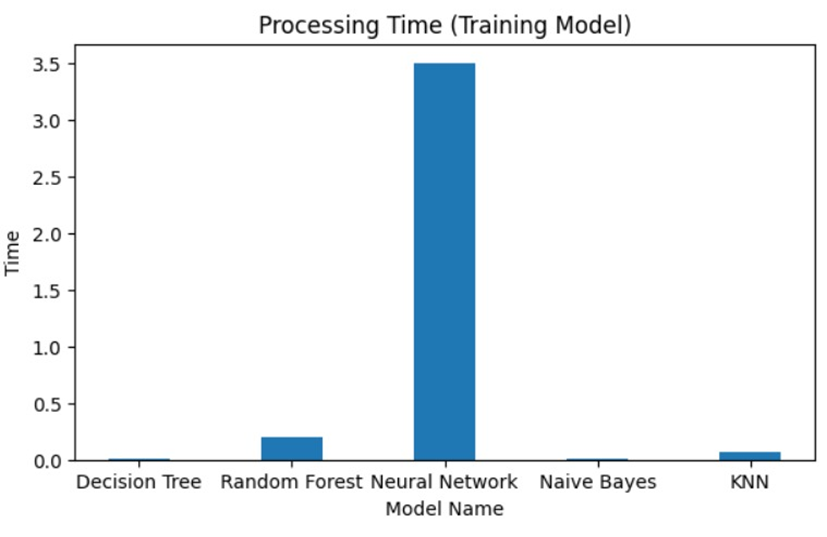


Figure 5 – Prediction Time for Machine Learning Models

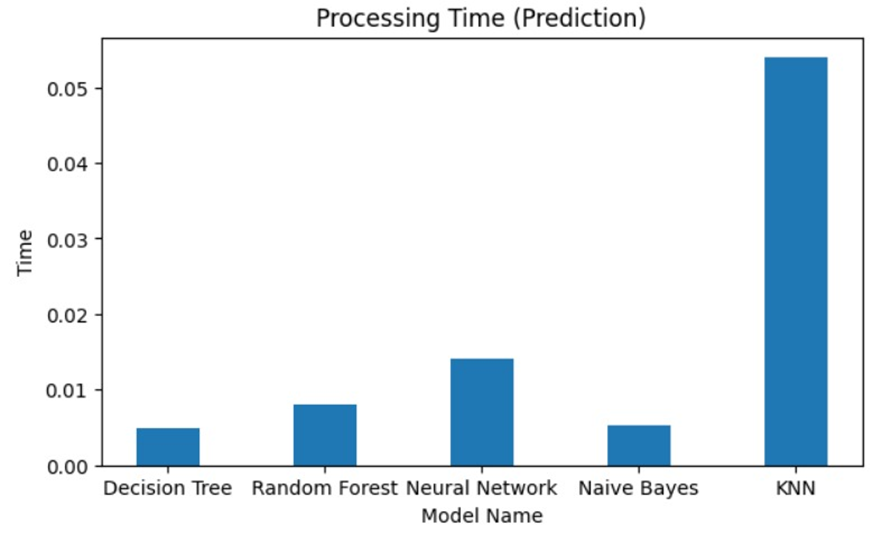


Figure 6 - TrainingTime for Machine Learning Models

**Results and Findings**

Our experimentation yielded promising results:

* We observed that Decision Tree outperformed Random Forest in terms of computational efficiency, making it the preferred choice for real-time detection in industrial environments.
* Feature selection significantly improved model performance, allowing us to focus on key features crucial for detecting brute force attacks while reducing computational overhead.

**Recommendations**

Based on our findings, we offer the following recommendations for enhancing intrusion detection systems:

* **Implement Decision Tree Model**: Deploy the Decision Tree model for real-time detection of brute force attacks in industrial environments due to its superior performance and computational efficiency.
* **Continuous Monitoring**: Implement continuous monitoring of network traffic using intrusion detection systems to promptly identify and mitigate brute force attacks as they occur.
* **Regular Updates and Training**: Regularly update and retrain intrusion detection models to adapt to evolving attack patterns and ensure optimal performance.
* **Collaborative Efforts**: Foster collaboration between cybersecurity experts, data scientists, and industry stakeholders to share insights and best practices in combating brute force attacks effectively.

**Conclusion**

In conclusion, our project demonstrates the efficacy of leveraging machine learning techniques to enhance intrusion detection systems, providing organizations with proactive measures to safeguard their critical systems and data against brute force attacks. By implementing our recommendations, organizations can fortify their cybersecurity posture and mitigate the risks posed by malicious actors in today's digital landscape.

# References

1. (PDF) Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset. (2020, November 1). ResearchGate. <https://www.researchgate.net/publication/360104704_Performance_analysis_of_intrusion_detection_for_deep_learning_model_based_on_CSE-CIC-IDS2018_dataset>
2. (PDF) SSH-Brute Force Attack Detection Model based on Deep Learning. (2023, February 1). ResearchGate. <https://www.researchgate.net/publication/348541012_SSH-Brute_Force_Attack_Detection_Model_based_on_Deep_Learning>
3. CanadianInstituteForCybersecurity. (n.d.). GitHub - CanadianInstituteForCybersecurity/CICFlowMeter: CICFlowmeter-V4.0 (formerly known as ISCXFlowMeter) is a network traffic Bi-flow generator and analyzer for anomaly detection that has been used in many Cybersecurity datsets such as Android Adware-General Malware dataset (CICAAGM2017), IPS/IDS dataset (CICIDS2017), Android Malware dataset (CICAndMal2017) and Distributed Denial of Service (CICDDoS2019). GitHub. <https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter>
4. Gautam, R.K.S.; Doegar, E.A. An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms. In Proceedings of the 2018 8th International Conference on Cloud Computing, Data Science & Engineering, Noida, India, 11–12 January 2018. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=An+Ensemble+Approach+for+Intrusion+Detection+System+Using+Machine+Learning+Algorithms&conference=Proceedings+of+the+2018+8th+International+Conference+on+Cloud+Computing,+Data+Science+&+Engineering&author=Gautam,+R.K.S.&author=Doegar,+E.A.&publication_year=2018)]
5. IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). <https://www.unb.ca/cic/datasets/ids-2018.html>
6. patator | Kali Linux Tools. (n.d.). Kali Linux. <https://www.kali.org/tools/patator/>
7. Songma, S., Sathuphan, T., & Pamutha, T. (2023). Optimizing intrusion detection systems in three phases on the CSE-CIC-IDS-2018 dataset. Computers, 12(12), 245. <https://doi.org/10.3390/computers12120245>
8. Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. Journal of Big Data, 11(1). <https://doi.org/10.1186/s40537-024-00886-w>