Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Random Forest-Based DDOS Detection from Cpanel Logs with Real-Time Notification Integration

Ridho Alfarizi[1]*, Akim M H Pardede[2], Husnul Khair[3]

[1,2,3]STMIK Kaputama
ridhoalfarizi.f@gmail.com[1*], akimmhp@gmail.com[2], husnul.khair@gmail.com[3]

Abstract

The study focuses on designing an automated program to detect Distributed Denial of Service (DDoS) attacks by analyzing access log data from CPanel. Using the Random Forest algorithm, the system processes large volumes of server log entries to distinguish between normal and malicious requests. Data preprocessing and model training are applied to optimize detection accuracy. To accelerate incident response, the detection module is integrated with Firebase Cloud Messaging (FCM), which delivers instant alerts to administrators when suspicious activity is identified. Experimental evaluation shows that the system achieves more than 95% accuracy on the test dataset, confirming its capability to reliably identify DDoS patterns. In comparison to manual analysis, the automated approach demonstrates superior speed, consistency, and operational efficiency, significantly reducing the time needed to recognize and respond to threats. The results indicate that combining machine learning-based detection with real-time notification is a practical and effective strategy for strengthening server security

Keywords: *Cybersecurity; DDoS Detection; Firebase Cloud Messaging; Machine Learning; Random Forest*

1. Introduction

The rapid development of information technology has led to an increase in internet-based services. However, this growth is accompanied by escalating cybersecurity threats[1], one of which is the Distributed Denial of Service (DDoS) attack. A DDoS attack aims to disrupt services by overwhelming a server with massive network traffic, thereby hindering access for legitimate users[2][3].

In server management, particularly those utilizing CPanel, access logs serve as a crucial source of information for monitoring network activity. Manual log analysis requires considerable time and effort, especially when dealing with hundreds of thousands of entries. This process poses the risk of delayed detection, reducing the effectiveness of threat mitigation.

Previous studies have explored the use of Machine Learning methods for cyberattack detection. However, the application in CPanel server environments and the integration with real-time notification systems remain limited, presenting opportunities for improvement[4][5][6].

This study aims to develop an automated DDoS attack detection system based on the Random Forest method, analyzing CPanel access log data to identify malicious activities. The system is enhanced with Firebase Cloud Messaging (FCM) integration to deliver instant alerts to administrators when an attack is detected, thereby accelerating response and mitigation. The main contribution of this research is to provide an effective, accurate, and responsive detection tool that enhances server security and service availability.

2. Literature Review

2.1. Machine Learning

Machine Learning is a branch of artificial intelligence that enables computer systems to learn from data without being explicitly programmed. This process involves processing input data to generate a model capable of making predictions or decisions. Learning is carried out by identifying patterns within the data, allowing the system to improve its performance over time[7].

2.2. Random Forest

Random Forest adalah metode ensemble learning yang digunakan untuk klasifikasi maupun regresi dengan membangun sejumlah pohon keputusan (decision tree) dan menggabungkan hasil prediksinya. Konsep utama dari metode ini adalah mengurangi risiko overfitting yang sering terjadi pada pohon keputusan tunggal dengan memanfaatkan banyak pohon yang dilatih pada subset data dan fitur yang berbeda[8].

2.3. Analysis

Analysis is a systematic process of breaking down data or information into smaller components to understand the structure, patterns, relationships, or meaning contained within. This process involves the critique, classification, and interpretation of data to answer research questions or test hypotheses[9]. Through careful and structured analysis, researchers can uncover hidden patterns, identify trends, and derive meaningful insights that form the basis for informed decision-making, hypothesis testing, and the development of effective strategies.

2.4. DDOS Attack

A Distributed Denial-of-Service (DDoS) attack is a type of cyberattack aimed at disrupting service availability by overwhelming the target system with excessive traffic from multiple sources that are centrally controlled by the attacker[6]. Mitigating DDoS attacks requires a combination of advanced detection mechanisms, real-time traffic monitoring, and proactive response strategies to minimize downtime and ensure that critical systems remain accessible even under high-volume attack conditions.

2.5. Cpanel

WHM/CPanel is a web hosting control panel on Linux that provides a graphical interface and optimization tools designed to simplify the website hosting process. In addition to offering a user-friendly graphical interface to facilitate user operations[10]. CPanel also provides comprehensive automation features, including automated backups, account management, and server maintenance tools, which significantly reduce administrative workload and allow web administrators to manage multiple websites efficiently.

2.6. Google Colaboration

This platform is developed based on Jupyter Notebooks technology. Jupyter is an open-source, browser-based software that integrates various programming languages, libraries, and visualization tools. Jupyter Notebooks can be run either locally or through cloud services[11]. Google Colaboratory further enhances this functionality by providing a cloud-based environment that allows multiple users to collaborate in real-time, access free GPU and TPU resources for accelerated computation, and seamlessly integrate with other Google services, making it an ideal platform for large-scale data analysis and machine learning projects.

2.7. Google Firebase

Firebase was first launched by Google to assist backend developers in managing their applications. With a variety of features offered, Firebase simplifies cloud database integration for both web and mobile applications. Firebase reduces configuration and setup complexity, making it more efficient and time-saving[12]. Beyond simplifying backend management, Firebase offers real-time data synchronization, built-in authentication mechanisms, performance monitoring, and analytics tools, providing developers with a complete ecosystem to build, scale, and maintain robust and interactive applications efficiently.

2.8. Android Studio

Android Studio is a software application developed as a tool to assist developers in building and designing Android-based systems[13]. It provides an integrated development environment (IDE) with features such as code editing, debugging, performance analysis, and emulator support, allowing developers to efficiently create, test, and deploy high-quality Android applications.

2.9. Kotlin

Kotlin adalah bahasa pemrograman yang berjalan di atas Java Virtual Machine (JVM) dan menggunakan kompiler LLVM, yang memungkinkan kode Kotlin dikompilasi menjadi JavaScript[13]. It offers modern language features, strong type inference, and full interoperability with Java, making it a preferred choice for Android development due to its concise syntax, safety features, and ease of maintenance.

3. Literature Review

3.1. Research Metodology

The approach taken in this study is quantitative research, involving a series of processes and procedures to analyze data obtained from a server's access logs. The development method used in this research is the Prototype Model of the Software Development Lifecycle (SDLC), selected for its flexibility while maintaining the quality of the tools being developed.

3.2. Model Implementation

In this study, the tools employed combine various aspects. The analysis process involves the Random Forest method as the primary approach, as illustrated in the flowchart below :

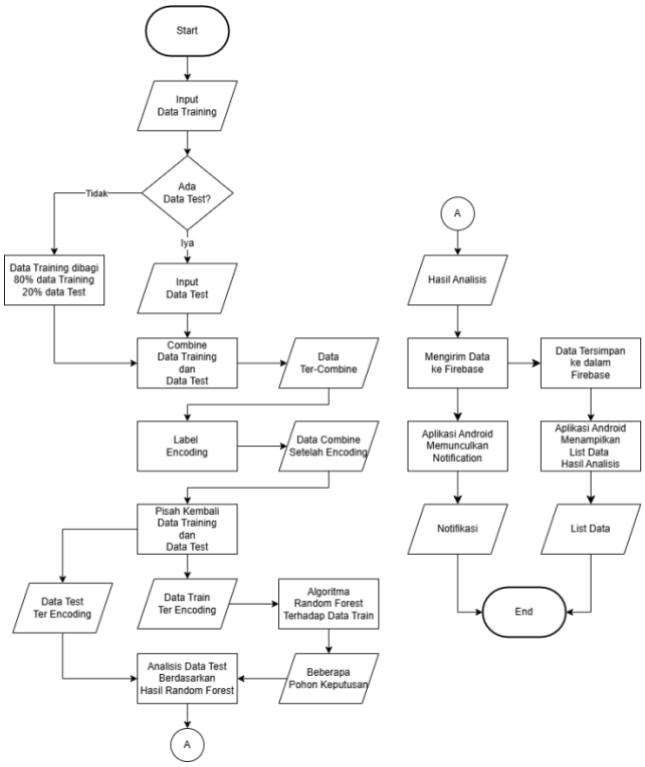

Fig. 1: Flowchart

Flowchart Explanation :

1. Start : The process begins with system initialization.
2. Input Training Data: The system receives training data to be used for building the model.
3. Check Availability of Test Data :
   a) If no test data is available, the training data is split into 80% training and 20% testing portions.
   b) If test data is available, the system proceeds directly to inputting the test data.
4. Input Test Data : The system receives the test data to be used for evaluating model performance.
5. Combine Training and Test Data : Both training and test datasets are merged for joint processing.
6. Label Encoding : The process of converting categorical data into numerical format so that it can be processed by the machine learning algorithm.
7. Re-split Training and Test Data : After encoding, the data is separated again according to its role (training or testing).
8. Process on Training Data :
   a) Encoded training data is used to train the model using the Random Forest algorithm.
   b) The algorithm generates multiple decision trees.
9. Process on Test Data :
   a) Encoded test data is analyzed using the trained Random Forest model.
   b) The system produces analysis results based on the model's predictions.
10. Send Analysis Results to Firebase : The analysis results are sent and stored in Firebase.
11. Android Application :
    a) Displays notifications when an attack is detected.
    b) Provides a list of analysis results within the application.
12. End : The process is completed.

3.3. Data Collection

In this study, the dataset used for analysis was obtained from the CPanel server access logs, consisting of nine variables: IP Address, Timestamp, Method Type, URL, Status Code, Response Size, Referer, User Agent, and Type. Not all variables were utilized as features or the target; therefore, a selection process was conducted to define the analysis scope. Five variables were chosen as features and one as the target variable for processing with the Random Forest method. This selection narrowed the analysis focus and enabled a deeper examination of DDoS attack patterns.

Table 1: Variable

Variable Name	Description
Method Type	Detects unusual spikes in specific HTTP request types.
URL	Identifies repeated or abnormal endpoint access patterns.
Status Code	Flags excessive error responses as potential attacks.
Referer	Highlights invalid or empty sources indicating bot activity.
User-Agent	Detects repeated or invalid client identifiers.
Type (Target)	label indicating normal or DDoS traffic for model training

After the access log data is collected from the CPanel server, from the nine initial variables, feature selection is performed to determine the most relevant attributes for DDoS detection. As a result, five variables — Method Type, URL, Status Code, Referer, and User-Agent — are used as features, while the Type variable serves as the target label for classification.

The dataset is then processed using the Random Forest algorithm. All selected features are encoded into numerical form through label encoding, enabling the algorithm to process categorical data efficiently. The dataset is divided into training and testing sets to evaluate model performance objectively. The Random Forest algorithm constructs multiple decision trees, each trained on random subsets of the data and features, to reduce overfitting and improve classification accuracy.

The trained model produces main outputs: the predicted class label indicating whether a record is normal or a potential DDoS attack. These results are stored in Firebase and integrated into a notification system via Firebase Cloud Messaging (FCM), allowing administrators to receive real-time alerts alongside a detailed list of detected events.

3.4. Android Interface

The application integrates with Firebase Cloud Messaging (FCM) to deliver push notifications directly to the administrator's mobile device. These notifications contain concise information, including the alert type, affected endpoint, and the detection timestamp, ensuring that the administrator can respond immediately without the need to constantly monitor the application. This dual notification approach—both in-app and push notifications—enhances situational awareness and supports timely mitigation actions.

main interface presents real-time alerts of detected DDoS attacks in a clear and structured format. Each alert card displays a warning icon, the affected URL endpoint, the number of times the website has been accessed, the number of detected attack attempts, and the latest update timestamp. This design allows administrators to quickly identify which resources are under attack and assess the frequency of malicious requests.

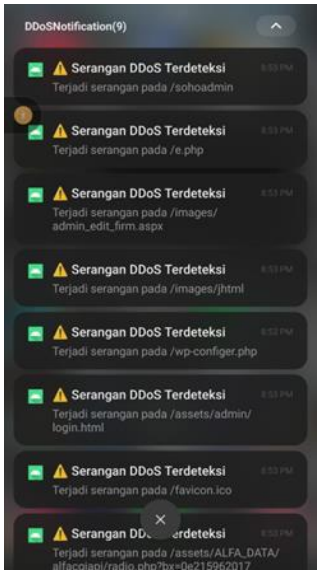

Fig. 1: Android Firebase Notification

Fig. 2: Android List Interface

4. Discussion And Implementation

4.1. Experimental Results

The testing was conducted to evaluate the performance of the Random Forest algorithm in accurately analyzing DDoS attacks and delivering notifications in a timely and responsive manner.

Three testing scenarios were carried out by varying the amount of data analyzed, the number of trees, and the tree depth. The results showed that the required processing time was shorter compared to manual analysis.

Table 2: Variable

Trial	Data Volume	Number of Trees	Depth	Accuracy	Execution Time
1	235,252	100	5	99%	6.6 minutes
2	117,626	50	3	98%	4.14 minutes
3	23,525	10	3	97%	52 seconds

Based on these testing results, datasets with smaller volumes and lower Random Forest specifications have shorter execution times, but with a slight compromise in accuracy. Considering the urgency of the analysis results, accuracy is deemed more important than execution time. Although execution time varies depending on the dataset size, it is still significantly faster compared to manual analysis.

4.2. Comparison

The comparison between the manual process and the implementation of the model can be seen in the following table :

Table 3: Comparison Manual and Automated Method

Aspect	Manual	Random Forest
Analysis Method	Observing network logs or traffic directly, calculating averages, traffic peaks, and comparing with the normal baseline.	Using a trained machine learning model to identify patterns of normal traffic vs. attacks.
Speed	Slow, as it requires manual observation and calculation per session or log.	Fast, only requires model execution time after training.
Accuracy	Depends on the analyst's expertise and the amount of data reviewed; prone to human error.	High (depending on training data quality), can reach >95% accuracy according to model results.
Data Requirement	Only requires part of the log data for direct analysis.	Requires a large labeled dataset (normal/attack).
Scalability	Difficult for large-scale traffic due to human limitations.	Can analyze millions of packets or requests in a short time.
Skill Requirement	Must master traffic analysis, network protocols, and anomaly detection techniques.	Must understand data processing, programming, and machine learning.

5. Conclusiona and Recommendations

5.1. Conclusion

This study successfully implemented the Random Forest method to analyze access patterns in CPanel logs, enabling automatic detection of DDoS attacks. Through data transformation and model training, the system was able to distinguish between valid access and attack traffic with high accuracy, achieving more than 95% accuracy rate in testing without manual intervention. The integration with Firebase Cloud Messaging (FCM) provided instant notifications to administrators upon detection of an attack, significantly reducing response time by enabling immediate awareness and mitigation, even when administrators were away from the server. Furthermore, the comparison results demonstrated that the automation tool is considerably faster, more consistent, and less prone to errors than manual analysis, which requires substantial time to review large volumes of log data. The automated system efficiently processes large-scale data and delivers detection results in a short time, thereby enhancing operational efficiency and minimizing the risk of delayed attack detection.

5.2. Recommendations

The recommendations for further development of this research are as follows:

1. To enhance security, the system can be directly integrated with a firewall or Web Application Firewall (WAF) so that when an attack is detected, IP blocking or termination of malicious access can be carried out automatically without waiting for administrator intervention.
2. In addition to notifications to Android devices, the system can be extended to send alerts via email, SMS, or a centralized monitoring dashboard. This way, threat information can be received more widely by the security team on duty.
3. Further testing on servers with real-time traffic and large data volumes is highly recommended to ensure the system's performance remains optimal and responsive under actual operational conditions.
4. Since DDoS attack patterns continuously evolve, the Machine Learning model needs to be retrained regularly using the latest datasets to remain relevant and accurate in detecting threats.

References

[1] L. Ikhwanul Uzlah and R. Adi Saputra, "Deteksi Serangan Siber Pada Jaringan Komputer Menggunakan Metode Random Forest," 2024.
[2] A. Harris and A. Rahim, "Seleksi Fitur dengan Information Gain untuk Meningkatkan Deteksi Serangan DDoS Menggunakan Random Forest An Information Gain Feature Selection to Improve DDoS Detection using Random Forest," 2020.
[3] B. Kashyap and S. K. Jena, "DDoS Attack Detection and Attacker Identification," 2012.
[4] F. Riza, "Sistem Deteksi Intrusi pada Server secara Realtime Menggunakan Seleksi Fitur dan Firebase Cloud Messaging," Jurnal Sistim Informasi Dan Teknologi, pp. 7–15, 2023.
[5] R. Ma, Q. Wang, X. Bu, and X. Chen, "Real-Time Detection of DDoS Attacks Based on Random Forest in SDN," Applied Sciences, vol. 13, no. 13, p. 7872, 2023.
[6] J. Pei, Y. Chen, and W. Ji, "A DDoS attack detection method based on machine learning," in Journal of Physics: Conference Series, IOP Publishing, 2019, p. 032040.
[7] I. D. Id, Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python, vol. 1. Unri Press, 2021.
[8] R. Genuer, J.-M. Poggi, R. Genuer, and J.-M. Poggi, Random forests. Springer, 2020.
[9] L. Cohen, L. Manion, and K. Morrison, "Research Methods in Education," 2018.
[10] M. N. Abdiansyah, Manajemen Hosting Berbasis WHM/cPanel. Excellent Publishing, 2018.
[11] D. S. R. Sukhdeve and S. S. Sukhdeve, "Google Colaboratory," in Google Cloud Platform for Data Science: A Crash Course on Big Data, Machine Learning, and Data Analytics Services, Springer, 2023, pp. 11–34.
[12] M. Tram, "Firebase," 2019.
[13] I. H. Hardy, E. C. Sujadi, and S. F. Pane, Pengembangan Smart Conveyor dengan Arduino (menggunakan GPS tracking berbasis android). Penerbit Buku Pedia, 2023.