An Advanced Traffic Analysis System for Detecting Web-Based Distributed Denial of Service Attacks

S. Vigneshwaran[1], Dr.S. Latha[2]
[1]*Student, Dr.MGR Educational and Research Institute, Chennai, India*
[2]*Director, Center for Cyber Forensics and Information Security, University of Madras, India*

***Abstract*—With the rapid growth of internet-based services, websites and web applications have become prime targets for cyberattacks such as Denial of Service (DoS) and Distributed Denial of Service (DDoS). These attacks can severely impact service availability, performance, and user trust. To address this challenge, this project presents a web-based DoS/DDoS Detection and IP Reputation Analysis System developed using modern web and cybersecurity technologies.**

The system is built using Python and Django as the backend framework, with HTML, CSS, Bootstrap, and JavaScript used for creating an interactive and responsive frontend interface. The application follows a modular architecture and includes User and Admin modules to ensure secure access control and effective system management.

Overall, this project serves as an effective cybersecurity monitoring and learning tool, combining traffic analysis, DoS/DDoS detection, IP reputation checking, and data visualization. It is suitable for academic, research, and small-scale real-world security monitoring applications, demonstrating practical implementation of network security concepts using modern web technologies.

***Index Terms*—Behavioral Analysis, DDoS Detection, IP Reputation, Threat Intelligence, Traffic Analysis.**

I. INTRODUCTION

With the increasing dependency on websites and online services, cyberattacks such as Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks have become major threats to organizations and web applications. These attacks attempt to overload servers with excessive requests, causing slow response times, service interruptions, or complete downtime. Detecting such abnormal traffic behavior at an early stage is important for maintaining website availability and security.

The proposed system is a web-based cybersecurity monitoring framework designed to analyze websites and detect possible DoS/DDoS behavior using repeated scanning and traffic analysis techniques. The system allows users to perform both quick and deep scans on target URLs and provides detailed information such as response latency, HTTP error codes, DNS details, geolocation data, and server response behavior. The project also integrates IP reputation analysis using the Abuse IPDB API to identify suspicious or malicious IP addresses based on global abuse reports.

The project is developed using Python and Django for backend processing and HTML, CSS, Bootstrap, and JavaScript for frontend design and visualization. The system is divided into User and Admin modules to ensure secure and organized management of the platform.

II. METHODOLOGY

2.1 Requirements & Architecture Design

- In Windows system used Python for programming and Django used for framework. Database has been built using SQLite. Libraries requests (HTTP requests), dnspython(DNS resolution), ipwhois(IP geolocation & ASN lookup), smtplib(Email sending) were mapped.
- Frontend has developed using HTML, CSS, Bootstrap, Javascript. API Integration was done with AbuseIPDB for CHECKIP provision.
- Storage perspective At least 20GB free disk space and decent processor. Stable internet connection API calling and scanning.

2.2 Access Control Implementation

- Implemented user registration with an email based One-time password verification using smtplib to validate user identity.
- Constructed a administrative dashboard allowing visibility and permissions into the active users.
- Admin has a authority to suspend user and reactivate the suspended users.

2.3 Traffic Scanning Development

This core module evaluates target web application performance parameters to identify active network degradation. The framework provides two distinct scanning modes:

- Quick Scan: This runs a single, fast check on the URL using the requests library. It looks at the website's basic server parameters and immediately displays response latency in milliseconds, any HTTP error codes (like 429 or 503), the server's response headers, and geolocation info. It also logs this data into the ScanResult table.
- Deep Scan: To actually spot a real DoS or DDoS attack, a single check isn't enough because server traffic fluctuates. The Deep Scan function sends a wave of exactly 20 repeated requests at fixed intervals to map traffic patterns over a period of time.

2.4 Threat Intelligence

To complement basic latency metrics, the proposed system cross-checks traffic origins via threat intelligence tools. The integration of a dedicated Threat Intelligence layer addresses this requirement.

When a scan is running, the system takes the target website's resolved IP address and queries the AbuseIPDB database using designated API key. The external API passes back live security parameters, which is subsequently stored in the AbuseReport table.

A live IP Blacklist interface was also implemented. This allows the user to see a clear "Abuse Confidence Score" scaled from 0 to 100, the ISP hosting the IP, its country code, and how many times this specific IP has been reported for malicious activity globally.

This feature pulls down a fresh list of the most highly reported malicious IPs from around the web, giving users an immediate look at active, high-risk digital threats without needing to scan them manually.

2.5 Verification

To verify if an attack is happening, To verify if an attack is occurring, the system architecture analyzes the metrics accumulated during the 20-loop Deep Scan. The system checks three main variables: Latency Spikes (sudden delays in server response times), Error Surges (a rapid jump in HTTP error replies), and Unreachable Periods (when a request completely times out).

If the average latency stays low and error counts are zero, the system classifies the website's behaviour as Normal. If response times jump drastically or timeouts add up, the system flags DOS_SUSPECTED or DDOS_SUSPECTED in the database, updating the behaviour status to Possible DoS or Possible DDoS.

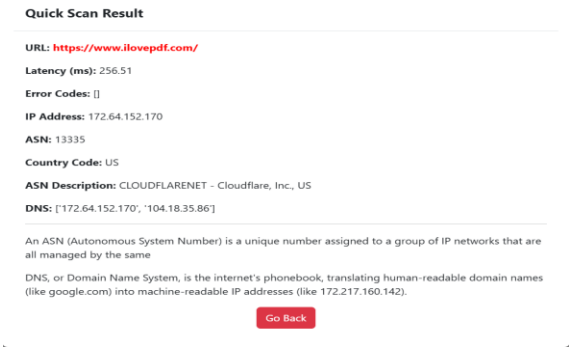
To enhance data readability, the JavaScript-based frontend transforms raw database logs into interactive, intuitive charts and visual graphs.

Users can also view their Scan History dashboard, which displays a clean archive of past scans, lets them open previous charts to check history trends, or delete old logs to keep their personal dashboard clean.

III. RESULTS AND VALIDATION

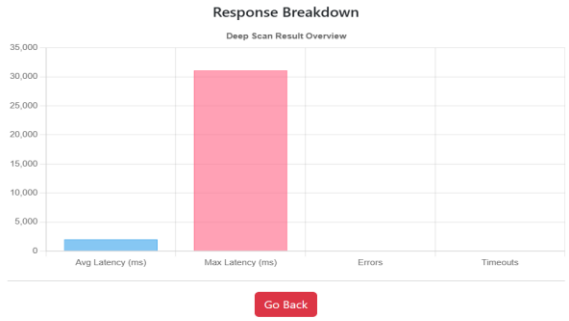
3.1 Live traffic scan analysis

The platform was tested by running live quick and deep scans on multiple target URLs to see how well it captures data and tracks server behaviour. During normal conditions, the Quick Scan safely retrieved server response headers, DNS routing addresses, and clean geolocations without any issues.



When running a Deep Scan with 20 repeated request loops, the system effectively mapped out performance differences. On standard, stable web apps, response latencies stayed uniformly low, resulting in an attack classification of Normal. However, when target

servers were deliberately subjected to heavy request loads, the frontend graph instantly highlighted dramatic latency spikes and frequent request timeouts.



3.2 IP Threat Metrics and System Suitability

The integration with the AbuseIPDB API successfully pulled back real-time reputation indicators for audited IP addresses. High-risk IPs triggered clear confidence scores close to 100%, warning the user of active malicious threats. The IP Blacklist view successfully populated an organized table showing the most recently reported malicious networks across different country domains.

To verify the overall efficiency and processing speed of the local software configuration, table logs were maintained to track database writes and API reply delays. The SQLite database updated user profiles and threat logs smoothly, showing zero system slowdowns even during the active 20-loop Deep Scan tracking.

IV. CONCLUSION

This web-based DoS/DDoS detection and IP reputation analysis system results demonstrate that highly capable network security monitoring solutions can be constructed without relying on cost-prohibitive, enterprise-grade tools. By combining simple tools like Python and Django with global intelligence endpoints like the AbuseIPDB API, the framework delivers a unified dashboard that facilitates cost-effective, centralized identification of DoS and DDoS anomalies. The system provides an accessible platform for developers and small-scale enterprises to execute traffic analysis without requiring complex network architecture configurations. The system provides an accessible platform for developers, small-scale enterprises, 20-loop traffic tests, track response graphs, and understand server safety patterns without requiring complex network architecture configurations. For the Future Scope, the system can be expanded much further. Right now, it relies on manual checks and simple numerical rules to spot anomalies. Moving forward, we plan to integrate Machine Learning algorithms so the application can automatically learn and identify complex, evolving attack signatures on its own. We also want to transition from manual checks to Real-Time Automated Monitoring, allowing the system to watch background live server logs constantly and send immediate alert notifications (like SMS or email messages) the exact second a traffic surge or botnet attack hits the system.

REFERENCES

[1] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 8th ed. Boston, MA, USA: Pearson, 2021.

[2] W. Stallings, Network Security Essentials: Applications and Standards, 7th ed. Boston, MA, USA: Pearson, 2017.

[3] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, pp. 39-53, Apr. 2004.

[4] OWASP, "OWASP Web Security Testing Guide," Version 4.2, 2023.

[5] AbuseIPDB, "AbuseIPDB API Documentation," 2026. [Online]. Available: https://docs.abuseipdb.com/. [Accessed: Jun. 11, 2026].

[6] Django Software Foundation, "Django Web Framework Documentation," 2026. [Online]. Available: https://docs.djangoproject.com/. [Accessed: Jun. 11, 2026].

[7] C. Douligeris and A. Mitrokotsa, "DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art," Computer Networks, vol. 44, no. 5, pp. 643-666, Apr. 2004.

[8] NIST, "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication 800-94, 2022.

[9] Python Software Foundation, "Python 3 Documentation," 2026. [Online]. Available: https://docs.python.org/3/. [Accessed: Jun. 11, 2026].

[10] Cloudflare, "Understanding and Mitigating DDoS Attacks," Cloudflare Learning Center, 2024.