IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

Ashtrashield A Hybrid ID (Intrusion Detection) for Network Protection

Pranav Deokar[1], Nikita Dhawale[2], Pratiksha Dhage[3], Aditya Devkar[4], Prof. L. S. Bhadake[5], Prof. N. M. Kolhe[6], Dr. H. B. Jadhav[7]
Student, Department of Computer Engineering[1234]
Faculty, Department of Computer Engineering[567]
Adsul's Technical Campus, Ahilyanagar, Maharashtra, India

Abstract: *Ashtrashield is a proposed lightweight security-monitoring platform designed to integrate web-application authentication, audit records, and host-level monitoring within a unified interface. The system architecture includes a frontend developed using HTML, CSS, JavaScript, and Bootstrap, PHP-based backend endpoints, and a MySQL database for managing user information, sessions, device records, and security logs. It also incorporates a proposed Python-based monitoring component using the psutil library to examine selected process, memory, and network indicators. The primary objective is to provide a centralized platform for managing application-related security information and initiating controlled host-monitoring operations. The system aims to improve visibility into application activities through structured logging and administrator-oriented monitoring workflows. However, the supplied project materials indicate that certain components, including the quickscan.py script and database schema/export, are unavailable. Furthermore, completed detection experiments and measured performance results have not been provided. Therefore, the proposed system is presented as an academic security-monitoring prototype, and its complete functionality requires further implementation, integration, and systematic testing.*

Keywords: Intrusion Detection, Host Monitoring, Web Application Security, PHP, MySQL, Python, psutil, Audit Logging.

I. INTRODUCTION

The rapid growth of computer networks and internet-based applications has increased the need for effective network security. Organizations and individuals rely on digital systems to store, process, and exchange important information. However, these systems are exposed to various security threats, including unauthorized access, suspicious activities, and malicious attempts to compromise system resources. Such threats can affect data confidentiality, integrity, and availability. Therefore, monitoring system activities and identifying potential security threats are important aspects of network protection.

An Intrusion Detection System (IDS) is a security mechanism designed to monitor network or system activities and identify events that may indicate unauthorized or suspicious behaviour. It helps administrators understand possible security incidents by analysing relevant activities and maintaining records of important events. Traditional intrusion detection approaches mainly focus on network traffic or host-level activities. Combining these monitoring perspectives can provide a broader view of system security.

Ashtrashield: A Hybrid ID (Intrusion Detection) for Network Protection is proposed as a lightweight securitymonitoring platform that combines web-application authentication and audit logging with planned host-level monitoring. The system uses a web-based interface developed using HTML, CSS, JavaScript, and Bootstrap. PHP is used for server-side operations, while MySQL is intended to store user details, device information, security logs, and related records. A Python-based monitoring component is planned to collect selected host-level indicators using the psutil library.

The proposed platform aims to provide a centralized approach to managing authentication-related activities and monitoring selected system information. By maintaining security logs and organizing monitoring records, the system is

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

intended to support the observation and review of potentially suspicious activities. The combination of web technologies, database management, and Python-based monitoring provides the foundation for the proposed hybrid approach.

This paper presents the motivation, problem definition, objectives, related work, system architecture, and methodology of Ashtrashield. It also discusses the current implementation status, security requirements, evaluation plan, limitations, and possible future improvements. The host-monitoring component requires further implementation and testing before its detection capabilities and performance can be validated.

II. PROBLEM DEFINITION AND OBJECTIVES

A. Problem Definition

The increasing use of web applications and computer networks has created a need for continuous monitoring and proper security management. Unauthorized access and suspicious system activities can create security risks. Without proper authentication mechanisms and activity records, it becomes difficult to monitor user access and review securityrelated events.

Many security-monitoring approaches focus on either network-level activities or host-level information. Managing these activities separately may limit the availability of a centralized view of security-related information. Therefore, a lightweight platform is proposed to combine web-application authentication, audit logging, and planned host-level monitoring.

The problem addressed by Ashtrashield is to develop a centralized security-monitoring platform that maintains authentication and audit records while providing a foundation for collecting selected host-level system information. The platform aims to support the observation and review of security-related activities through a web-based interface.

B. Objectives

The main objectives of the proposed Ashtrashield system are:

1. To develop a web-based interface for accessing and managing the security-monitoring platform.
2. To implement user authentication and session-based access management.
3. To maintain records of user activities and security-related events using a MySQL database.
4. To manage device information and related monitoring records.
5. To develop a Python-based component for collecting selected host-level system indicators using the psutil library.
6. To integrate web-application records with host-level monitoring as part of a hybrid security-monitoring approach.
7. To provide a foundation for future analysis and identification of potentially suspicious activities.

III. RELATED WORK

Intrusion Detection Systems (IDS) have been widely studied as an important component of computer and network security. Several research works have proposed different approaches for identifying unauthorized access, detecting suspicious activities, and monitoring system behaviour. The following studies provide the background for the proposed Ashtrashield system.

A. An Intrusion-Detection Model

Denning [1] proposed a general model for intrusion detection based on monitoring system activities and identifying abnormal behaviour. The model established the foundation for detecting security violations through the analysis of audit records. It highlighted the importance of monitoring user and system activities for identifying potential intrusions.

B. Guide to Intrusion Detection and Prevention Systems

Scarfone and Mell [2] presented guidelines for Intrusion Detection and Prevention Systems (IDPS) in NIST Special Publication 800-94. The publication explains different types of intrusion detection technologies, their deployment

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

methods, and their capabilities. It also discusses important considerations for selecting and managing securitymonitoring systems.

C. Machine Learning for Network Intrusion Detection

Sommer and Paxson [3] discussed the challenges of applying machine learning techniques to network intrusion detection. Their work highlighted the difficulties associated with real-world network environments, including changing traffic patterns and the limitations of available training data. It emphasizes that machine-learning-based detection requires careful evaluation before practical deployment.

D. Data Mining and Machine Learning Methods for Cyber Security

Buczak and Guven [4] presented a survey of data-mining and machine-learning methods used in cybersecurity intrusion detection. The study reviewed different techniques and their applications in identifying malicious activities. It provides an overview of how data-driven methods can support intrusion detection.

E. Survey of Intrusion Detection Systems

Khraisat et al. [5] reviewed intrusion detection techniques, datasets, and challenges. Their research discussed different IDS approaches and highlighted issues related to detection accuracy, false alarms, and the availability of suitable datasets. The study provides useful background for understanding the limitations and evaluation requirements of intrusion detection systems.

The reviewed studies establish the importance of activity monitoring, audit analysis, and systematic evaluation in intrusion detection. Based on this background, Ashtrashield proposes a lightweight approach that combines webapplication authentication and audit logging with planned host-level monitoring.

IV. SYSTEM ARCHITECTURE

The proposed Ashtrashield system follows a hybrid architecture that combines web-application operations with a planned host-level monitoring component. The architecture is designed to manage user authentication, maintain security-related records, and provide a foundation for monitoring selected system activities.

The system consists of the following major components:

A. User Interface

The user interface provides the interaction between the user and the Ashtrashield platform. It is developed using HTML, CSS, JavaScript, and Bootstrap. It provides web pages through which users can access the available application features.

B. PHP Server-Side Component

PHP is used to handle server-side operations and process requests received from the web interface. It supports application functions such as authentication, session handling, device records, security logs, and related data operations.

C. MySQL Database

MySQL is intended to store and manage the platform's application data. The planned database records include user information, device details, security logs, and related plan or payment records. It provides the data-storage foundation for the web application.

D. Python-Based Monitoring Component

A Python-based component is planned for host-level monitoring. The proposed implementation uses the psutil library to collect selected system indicators, such as process information, memory usage, and network-related information. This component is intended to extend monitoring beyond web-application activity.

E. Integration of Components

The web interface, PHP server-side component, and MySQL database form the main application structure. The Python monitoring component is planned as an additional part of the architecture. Their integration is intended to provide a combined view of application-related records and selected host-level information.

Suggested Architecture Diagram

Place the following diagram after the architecture description, if you are creating it separately:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

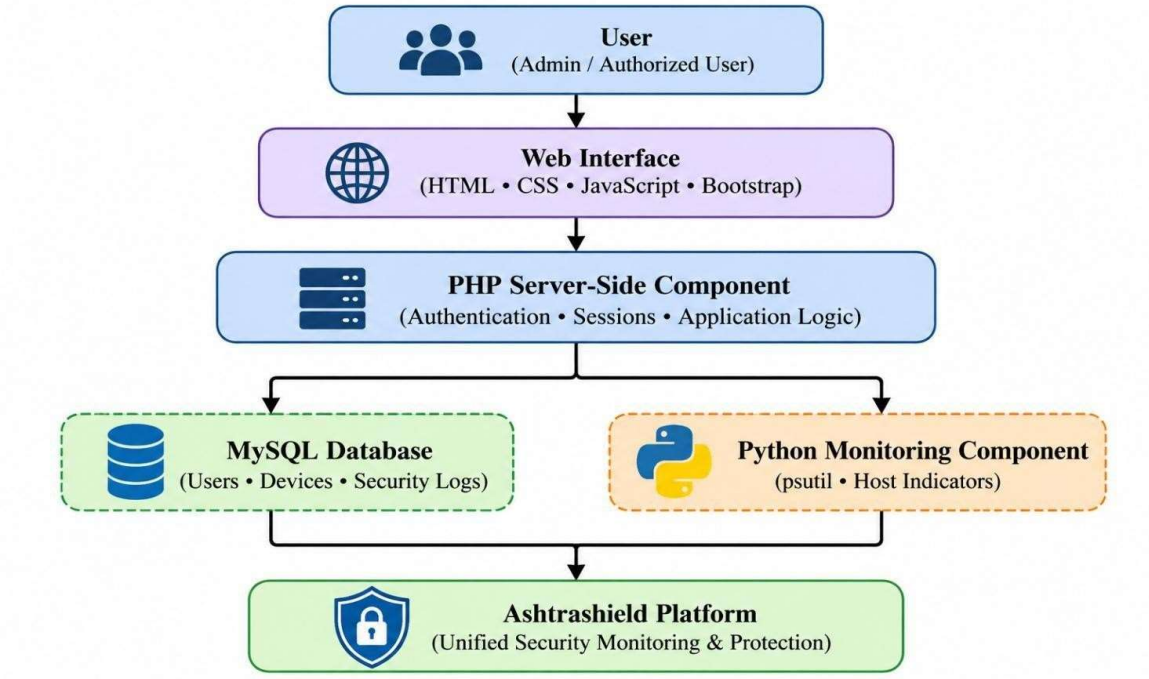


Fig. 1. Proposed System Architecture of Ashtrashield

V. METHODOLOGY AND REQUEST PROCESSING

The proposed Ashtrashield system follows a structured request-processing methodology in which the web interface communicates with the PHP server-side component. The server-side component processes application requests and interacts with the MySQL database to manage relevant information. A Python-based monitoring component is also planned to collect selected host-level indicators.

A. User Request

The process begins when a user accesses the Ashtrashield web application through a browser. The user interacts with the interface developed using HTML, CSS, JavaScript, and Bootstrap. The interface provides access to the available application functions.

B. Authentication and Session Management

When a user submits login information, the request is sent to the PHP server-side component for processing. The authentication process checks the submitted credentials against the stored user information. Session management is used to maintain the user's authenticated state during application access.

C. Database Operations

After processing a request, the PHP component communicates with the MySQL database to retrieve or manage the required information. The database is intended to maintain user details, device records, security logs, and other application-related records. These operations support the storage and retrieval of information required by the platform.

D. Security Logging

Security-related activities are intended to be recorded in the system's security logs. These records can help administrators review application activities and maintain an audit trail. The logging mechanism provides a basis for examining events associated with application access and operations.

E. Host-Level Monitoring

The proposed host-level monitoring component uses Python and the psutil library to collect selected system indicators, including process information, memory usage, and network-related information. The collected information is intended to

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

support system monitoring. The source paper indicates that the monitoring script, quickscan.py, is not available in the current implementation; therefore, this functionality requires further implementation and testing.

F. Request Processing Flow

The overall request-processing sequence can be described as follows:

1. The user accesses the Ashtrashield web application.
2. The web interface sends the user's request to the PHP server-side component.
3. The PHP component processes the request and performs the required authentication or application operation.
4. The PHP component interacts with the MySQL database to retrieve or store relevant records.
5. Security-related activities are maintained in the audit logs.
6. The planned Python monitoring component collects selected host-level indicators.
7. The stored application records and monitoring information provide a foundation for centralized security review.

VI. IMPLEMENTATION STATUS AND SECURITY REQUIREMENTS

The Ashtrashield system is designed using web technologies and a database-supported architecture. Its implementation focuses on user authentication, application request processing, device information management, and security logging. A Python-based component is planned to extend the system with host-level monitoring capabilities.

A. Technologies Used

The proposed system uses the following technologies:

1. **HTML:** Used to create the structure of the web pages.
2. **CSS:** Used to design and style the application interface.
3. **JavaScript:** Used to provide client-side interactivity.
4. **Bootstrap:** Used to develop a responsive user interface.
5. **PHP:** Used for server-side request processing, authentication, and database operations.
6. **MySQL:** Used to store user information, device records, and security-related logs.
7. **Python:** Planned for host-level monitoring operations.
8. **psutil:** Intended to collect system information, including process, memory, and network-related indicators.

B. Implemented Application Components

The available project structure describes PHP endpoints for handling authentication, session-related operations, device records, security logs, plan and payment records, and quick-scan requests. The MySQL database is intended to support the storage and management of application information.

The web application provides the foundation for managing user-related information and recording security events. These components are intended to support the overall monitoring workflow.

C. Current Implementation Status

The current project materials describe the web application and its server-side operations. However, the Python monitoring script named quickscan.py is not available in the supplied project material. The database schema and exported database structure are also not available for verification.

Therefore, the complete integration of host-level monitoring with the web application cannot yet be confirmed. The proposed monitoring functionality requires further development, integration, and testing before its effectiveness can be evaluated.

D. Security Requirements

To improve the security and reliability of Ashtrashield, the following requirements should be considered:

1. **Secure Authentication:** User credentials should be verified securely, and passwords should be stored using appropriate password hashing techniques.
2. **Prepared Statements**: Database queries should use prepared statements to reduce the risk of SQL injection.
3. **Session Security**: Sessions should be regenerated after successful authentication and invalidated during logout.
4. **Access Control:** Authentication and authorization checks should be applied to all protected server-side

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

endpoints.

5. **Secure Communication:** HTTPS should be used to protect information exchanged between the browser and server.
6. **Credential Protection:** Database credentials and other sensitive configuration details should be stored outside publicly accessible directories.
7. **Input Validation:** User inputs should be validated on the server side before processing.
8. **Log Protection:** Security logs should be protected from unauthorized access and should not contain passwords or session tokens.
9. **Monitoring Restrictions:** Any future scanning functionality should use restricted permissions, controlled execution time, and safe output handling.

These requirements are important for developing a reliable security-monitoring platform and reducing common webapplication vulnerabilities.

VII. EVALUATION PLAN

The evaluation of Ashtrashield is intended to verify the correctness of its application functions, authentication process, database operations, and planned monitoring capabilities. Testing is necessary to identify errors, evaluate system behaviour, and ensure that the application handles different situations appropriately.

Since the complete Python monitoring component is not available in the supplied project material, the following test cases represent a proposed evaluation plan. They should be executed after the corresponding functionalities are implemented.

A. Functional Testing

The following test cases are proposed to evaluate the main application functions.

TABLE I. PROPOSED FUNCTIONAL TEST CASES

Test ID	Test Scenario	Expected Result
T1	Login using valid credentials	User should be authenticated successfully.
T2	Login using invalid credentials	Access should be denied with an appropriate error message.
T3	Access a protected endpoint without a valid session	Request should be rejected.
T4	Normal user attempts to access an administrator function	Unauthorized access should be denied.
T5	Database connection is unavailable	Application should handle the failure safely.
T6	Python monitoring script is available	Monitoring request should execute within configured limits.
T7	Monitoring script is missing or exceeds the time limit	Application should handle the failure without exposing sensitive details.
T8	Submit malformed or unexpected input	Invalid input should be rejected safely.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

B. Performance Evaluation

Performance evaluation is planned to measure how efficiently the system processes application requests and manages system resources. The following parameters can be considered:

1. **Response Time:** Time required by the application to respond to a user request.
2. **Database Query Time:** Time taken to execute database operations.
3. **Monitoring Execution Time:** Time required by the Python monitoring component to collect system information.
4. **CPU Utilization:** Amount of processor resources consumed during monitoring operations.
5. **Memory Utilization**: Amount of memory used by the application and monitoring process.
6. **Record Processing:** Number of application or monitoring records processed during a defined period.

C. Detection Evaluation

After implementing the intrusion detection mechanism, its effectiveness can be evaluated using a labelled dataset containing normal and suspicious activities. The following evaluation metrics may be used:

- **Accuracy:** Measures the proportion of correctly classified instances.
- **Precision:** Measures how many activities identified as suspicious are actually suspicious.
- **Recall:** Measures how many actual suspicious activities are correctly identified.
- **False Positive Rate:** Measures how many normal activities are incorrectly classified as suspicious.

These metrics will help evaluate the reliability of the detection mechanism. They should only be reported after executing the detection system on a suitable test dataset.

D. Testing Documentation

For each test case, the testing process should record the test environment, input data, expected result, observed result, and final status. Screenshots, application logs, and error details can also be maintained as supporting evidence.

The evaluation plan provides a structured approach for testing the functionality, performance, and security-related behaviour of Ashtrashield. Actual test results must be recorded after the implementation and execution of the proposed test cases.

VIII. DISCUSSION

Ashtrashield presents a hybrid approach to security monitoring by combining web-application authentication and audit logging with planned host-level monitoring. The proposed architecture uses commonly available technologies such as HTML, CSS, JavaScript, Bootstrap, PHP, MySQL, and Python. This combination provides a foundation for developing a lightweight and manageable security-monitoring platform.

One of the important aspects of the proposed system is the integration of authentication and security logging. Authentication helps control access to the application, while audit logs provide records of relevant application activities. These records can support administrators in reviewing user access and investigating security-related events.

The planned Python-based monitoring component is another important part of the proposed architecture. By using the psutil library, the system is intended to collect selected host-level information, such as running processes, memory utilization, and network-related indicators. Combining this information with application-level records may provide a broader understanding of system activities.

The use of PHP and MySQL supports server-side request processing and structured data management. Additionally, the web-based interface provides a convenient way to interact with the application. These technologies make the proposed system suitable as a foundation for further development and integration.

However, the effectiveness of the hybrid approach depends on the successful implementation and integration of all components. The absence of the quickscan.py monitoring script and the lack of verified evaluation results mean that the host-level monitoring and intrusion detection capabilities cannot yet be considered validated. Further testing is necessary

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

to assess system performance, detection accuracy, and reliability. Overall, Ashtrashield provides a proposed framework for combining web-application security records with host-level monitoring. Its future effectiveness will depend on completing the monitoring component, strengthening security controls, and conducting systematic evaluation.

IX. LIMITATIONS AND FUTURE WORK

Although Ashtrashield proposes a hybrid approach to security monitoring, the current project has certain limitations that need to be addressed before it can be considered a complete intrusion detection solution.

A. Limitations

1. **Incomplete Monitoring Component:** The Python monitoring script, quickscan.py, is not available in the supplied project material. Therefore, host-level monitoring functionality cannot be fully verified.
2. **Database Structure:** The complete MySQL database schema and exported database structure are not available for verification.
3. **Limited Detection Capability**: The proposed system does not yet provide verified results from an implemented intrusion detection algorithm.
4. **Performance Evaluation:** Actual performance measurements, such as response time, CPU utilization, and memory consumption, have not been reported.
5. **Security Improvements:** Additional security measures, including prepared statements, strong session management, and server-side authorization checks, require implementation and verification.

B. Future Work

The following improvements can be considered in future development:

1. **Complete Host Monitoring:** Implement the Python monitoring script using psutil to collect process, memory, and network-related information.
2. **Intrusion Detection Mechanism:** Develop suitable rule-based or machine-learning-based techniques to identify suspicious system and network activities.
3. **Real-Time Monitoring**: Introduce periodic or real-time monitoring with appropriate alert mechanisms.
4. **Database Integration:** Complete the database schema and establish reliable communication between the web application and monitoring component.
5. **Dashboard Enhancement:** Develop a centralized dashboard to display security logs, monitoring information, and system status.
6. **Security Enhancement:** Implement secure authentication, password hashing, prepared statements, role-based access control, and protected log storage.
7. **Performance Optimization**: Improve request processing and monitoring efficiency to reduce unnecessary resource consumption.
8. **Comprehensive Testing:** Evaluate the system using suitable datasets and test cases to measure detection accuracy, precision, recall, false positive rate, and overall performance.

These improvements can help transform the proposed platform into a more complete and reliable security-monitoring solution. Further development and systematic testing will be necessary to establish its practical effectiveness in network protection.

Examples of reference items of different categories shown in the References section include:
example of a book in [1] example of a book in a series in
[2] example of a journal article in [3]

X. CONCLUSION

Ashtrashield: A Hybrid ID (Intrusion Detection) for Network Protection presents a proposed lightweight securitymonitoring platform that combines web-application authentication and audit logging with planned host-level monitoring. The system architecture uses HTML, CSS, JavaScript, and Bootstrap for the frontend, PHP for server-side

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

request processing, and MySQL for application data management. A Python-based monitoring component using the psutil library is proposed to collect selected host-level information, including process, memory, and network-related indicators.

The proposed approach aims to provide a foundation for managing application access, maintaining security records, and extending monitoring to host-level activities. By combining these components, Ashtrashield seeks to support a more organized approach to reviewing security-related information.

However, the complete implementation and integration of the Python monitoring component, along with comprehensive functional and performance testing, remain necessary. Therefore, the system's intrusion detection effectiveness and practical performance have not yet been experimentally validated.

In conclusion, Ashtrashield provides a foundation for developing a hybrid security-monitoring solution. Future improvements in host-level monitoring, intrusion detection techniques, security controls, and systematic evaluation can help enhance its reliability and usefulness for network protection.

XI. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the faculty members of the Department of Computer Engineering, Adsul's Technical Campus, Chas, Ahilyanagar, Maharashtra, for their valuable guidance and support throughout the development of this work.

The authors are thankful to their project guides and mentors for their suggestions, technical assistance, and encouragement. They also acknowledge the support of their institution, colleagues, and everyone who contributed directly or indirectly to this work.

REFERENCES

[1]. D. E. Denning, "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, Feb. 1987, doi: 10.1109/TSE.1987.232894.

[2]. K. Scarfone and P. Mell, Guide to Intrusion Detection and Prevention Systems (IDPS), NIST Special Publication 800-94, National Institute of Standards and Technology, 2007, doi: 10.6028/NIST.SP.800-94.

[3]. R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proceedings of the IEEE Symposium on Security and Privacy, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[4]. A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[5]. A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," Cybersecurity, vol. 2, Art. no. 20, 2019, doi: 10.1186/s42400-0190038-7.