AI Powered Network Intrusion Detection System (N-IDS) “FUSION SHIELD”

Tanmay R. Shrimali[1], Nakul P. Jadhav[2], Prathmesh D. Shinde[3], Asad N. Shaikh[4],
Dr. Monika S. Deshmukh[5]

[1,2,3,4] *Student, Department of Computer Science and Engineering in Cyber Security and Forensics, School of Computer Science and Engineering, Sandip University, Nashik, Maharashtra, India*

[5] *Assistant Professor, Department of Computer Science and Engineering, School of Computer Science and Engineering, Sandip University, Nashik, Maharashtra, India*

ABSTRACT

The fast-changing nature of cyber threats requires new ways to protect networks. This project focuses on creating an AI-powered Network Intrusion Detection System (NIDS) to improve traditional intrusion detection systems. By using machine learning (ML) and natural language processing (NLP), our NIDS aims to spot and handle potential intrusions in real-time.

Our NIDS works by collecting and studying network logs from endpoint devices using a special log forwarder agent. These logs go to a centralized Security Information and Event Management (SIEM) server for detailed analysis with ML algorithms. By integrating with the OpenAI GPT-3 API, our system can offer detailed insights and useful information about detected intrusions.

Key features of our NIDS include spotting anomalies in real-time, monitoring endpoints actively using custom Python scripts, and using AI to analyze and create detailed intrusion alerts. This report explains the design, implementation, challenges faced, and results achieved while making our AI-powered NIDS, showing how AI can change network security.

Our project helps improve cybersecurity by showing how AI can adaptively detect and respond to threats. The results emphasize the importance of using AI in intrusion detection systems to fight ever-changing cyber threats effectively.

Keyword: - *AI-powered N-IDS, Network Intrusion Detection System, Artificial Intelligence, Machine Learning, Deep Learning, Anomaly Detection, Cybersecurity, Threat Detection*

1. Introduction

Today, cybersecurity threats are constantly changing, creating big challenges for organizations trying to protect their important data and systems. With networks becoming more connected and cyber attacks getting more sophisticated, it's crucial to have strong intrusion detection and prevention systems. This project focuses on developing an AI-powered Network Intrusion Detection System (NIDS) to tackle these challenges.

The aim of this project is to create an advanced network security solution that can find and deal with potential intrusions as they happen. Traditional intrusion detection systems often use set rules and patterns, which might not be enough to handle modern, ever-changing threats. By using artificial intelligence (AI) technologies like machine learning (ML) and natural language processing (NLP), our NIDS can adapt and respond quickly to new threats.

Our NIDS focuses on analyzing logs and spotting anomalies. We install a log forwarder agent on endpoint devices to collect and send network logs to a central Security Information and Event Management (SIEM) server. These logs are carefully analyzed using machine learning to find patterns that might indicate malicious activities.

A big part of our NIDS is using a language model, specifically the OpenAI GPT-3 API. This lets us generate detailed insights and useful information when intrusions are detected. The system can summarize what happened during the intrusion, when it happened, and suggest ways to fix it, helping security analysts make quick decisions.

1.1 Overview

The "AI-Powered Network Intrusion Detection System" project aims to create a cutting-edge cybersecurity solution using artificial intelligence (AI) to boost network security and threat detection abilities. Given the growing complexity of cyber threats, traditional intrusion detection systems often struggle to effectively identify and respond to modern attacks. This project aims to overcome these challenges by integrating machine learning (ML) and natural language processing (NLP) into a comprehensive Network Intrusion Detection System (NIDS).

The project focuses on proactive threat detection and response by analyzing network logs and monitoring endpoints. We deploy a specialized log forwarder agent on endpoint devices to gather and send network logs to a centralized Security Information and Event Management (SIEM) server. These logs are then analyzed in real-time using ML algorithms to detect anomalies and potential intrusions.

A significant innovation of this project is the use of the OpenAI GPT-3 API to enhance intrusion analysis and response. This language model provides detailed insights into detected intrusions, including the type of threat, timelines of events, and recommended mitigation strategies. This AI-driven approach empowers security analysts to make informed decisions quickly, strengthening overall network security.

Furthermore, the project involves developing custom Python scripts for endpoint monitoring, allowing proactive scanning for open ports and vulnerabilities.

The main goal of this project is to demonstrate the potential of AI in transforming network security practices. By combining advanced ML techniques with real-time log analysis and endpoint monitoring, our AI-powered NIDS showcases a proactive and adaptive approach to intrusion detection, contributing to the continuous evolution of cybersecurity strategies.

1.2 Problem Definition

The project aims to overcome the limitations of traditional intrusion detection systems (IDS) in combating modern cyber threats effectively. Conventional IDS systems often use static rule-based methods and predefined signatures, which may not be enough to detect and respond to dynamic and evolving intrusion techniques. Therefore, there is a crucial need for more adaptive and intelligent intrusion detection mechanisms capable of proactively identifying and mitigating network intrusions.

Key challenges with current intrusion detection systems include:

1. **Inability to Detect Unknown Threats:** Traditional IDS struggle to identify unknown or zero-day threats that lack predefined signatures, leaving networks vulnerable to new attack methods.

2. **High False Positive Rates:** Rule-based IDS approaches often produce many false alarms, overwhelming security analysts and decreasing the effectiveness of incident response efforts.

3. **Limited Contextual Understanding:** Existing IDS systems may not fully understand the context of network activities and user behaviors, resulting in incomplete threat detection and analysis.

4. **Inefficient Incident Response:** Manual analysis required to interpret IDS alerts can lead to delayed or inadequate incident responses, prolonging exposure to security risks.

To tackle these challenges, this project aims to develop an AI-powered Network Intrusion Detection System (NIDS) that uses machine learning (ML) and natural language processing (NLP) techniques. The goal is to improve threat detection accuracy, reduce false positives, and provide comprehensive insights for effective incident response. By integrating AI into the NIDS, we aim to enhance the overall resilience of organizational networks against sophisticated cyber attacks and emerging threats.

2. Project Scope

This project aims to develop an AI-powered Network Intrusion Detection System (NIDS) to improve network security and threat detection capabilities. The project will involve the following key components and activities:

1. **System Design and Architecture:** Define the architecture and design principles of the AI-powered NIDS. This includes outlining components for log collection, real-time analysis, machine learning integration, and reporting.

2. **Data Collection and Preprocessing:** Implement a log forwarder agent to collect network logs from endpoint devices. These logs will be transmitted to a centralized Security Information and Event Management (SIEM) server for preprocessing and analysis.

3. **Machine Learning Model Development:** Develop and train machine learning models for anomaly detection and intrusion classification. This involves using network logs and historical data to explore supervised learning, unsupervised learning, and ensemble methods for effective threat detection.

4. **Integration of AI Technologies:** Integrate natural language processing (NLP) capabilities using the OpenAI GPT-3 API. This will help generate detailed insights and actionable intelligence about detected intrusions.

5. **Endpoint Monitoring and Scripting:** Develop custom Python scripts to perform regular scans of endpoint devices. These scripts will check for open ports, vulnerabilities, and suspicious activities, enhancing proactive threat detection and response.

6. **Real-time Monitoring and Alerting:** Implement real-time monitoring of network traffic and endpoint activities. This will enable prompt detection and response to potential security incidents. Configure alerting mechanisms based on predefined thresholds and rules for security analysts.

7. **User Interface and Dashboard:** Design and develop a user-friendly dashboard. This dashboard will visualize intrusion alerts, system status, and AI-generated insights for security analysts and stakeholders.

By completing these components and activities, the AI-powered NIDS will offer enhanced network security through proactive threat detection, real-time monitoring, and detailed analysis supported by AI technologies.

2.1 Architecture Overview

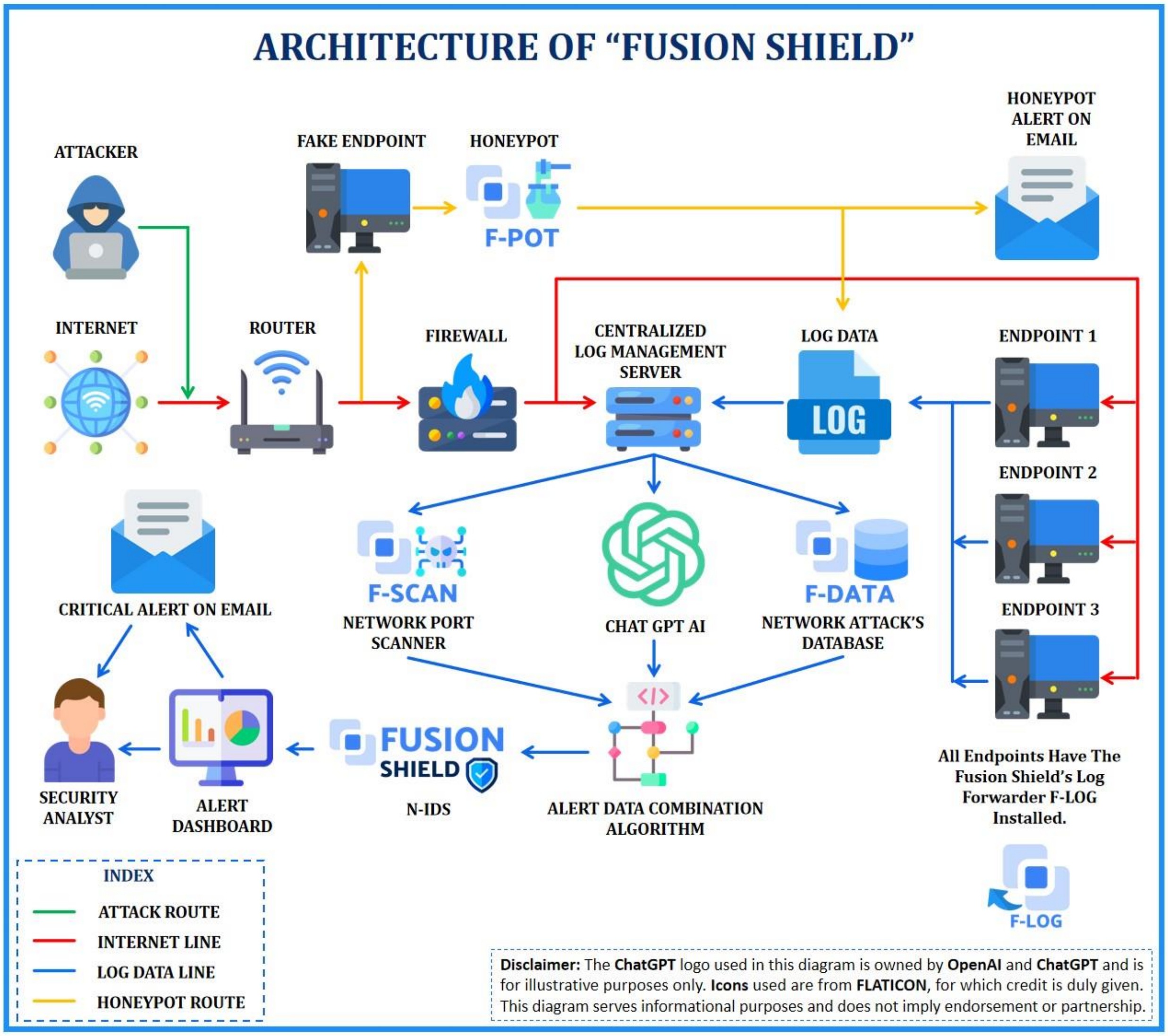
1. **Log Forwarder Agents:** Installed on endpoint devices to gather and send network logs to a centralized SIEM server.

2. **SIEM Server:** Centralized server responsible for receiving, preprocessing, and storing network logs.

3. **Machine Learning Module:** Integrated with the SIEM server to perform real-time log analysis, anomaly detection, and threat classification using machine learning.

4. **OpenAI GPT-3 Integration:** API integration to generate detailed insights and actionable intelligence on detected intrusions.

5. **Custom Python Scripts:** Developed for proactive endpoint monitoring, including vulnerability scans and security checks.



All Endpoints Have The Fusion Shield's Log Forwarder F-LOG Installed.

Disclaimer: The **ChatGPT** logo used in this diagram is owned by **OpenAI** and **ChatGPT** and is for illustrative purposes only. **Icons** used are from **FLATICON**, for which credit is duly given. This diagram serves informational purposes and does not imply endorsement or partnership.

Fig -1 Architecture Of Fusion Shield N-IDS

2. Implementation Of Project

In our project implementation, we have established a robust Security Information and Event Management (SIEM) server environment on Ubuntu, utilizing the ELK stack (Elasticsearch, Logstash, Kibana) as our core components. The ELK stack serves as the foundational platform for centralized log management and analysis.

To streamline log collection and analysis, we have deployed OSSEC agents as log forwarders on our endpoint devices. These agents systematically gather network logs and transmit them to our ELK server, enabling centralized storage and comprehensive analysis of network activities.

Moreover, we have developed a specialized Python script named F-SCAN. This custom script facilitates direct network scans on endpoint devices, allowing us to identify network vulnerabilities and gather critical network information in real-time.

A notable innovation in our SIEM setup is the integration of ChatGPT. Unlike conventional SIEM systems that primarily generate alerts, our system leverage's ChatGPT to enrich alert notifications with detailed descriptions and recommended mitigation steps. This integration significantly enhances our incident response capabilities by providing security analysts with actionable insights alongside alert notifications.

By integrating these components, we have established an advanced SIEM solution that not only detects security incidents but also empowers security analysts with contextual information and actionable guidance to respond effectively to threats. This holistic approach to SIEM leverages AI-driven insights to enhance security monitoring and incident response capabilities, ensuring a proactive and adaptive defence against evolving cybersecurity threats.

3. Network Scanner F-SCAN Algorithm

1. Argument Parsing: Parse command-line arguments using argparse to determine the scan configuration.

2. Port List Preparation: Based on the provided arguments (--ports, --range, or --common), prepare a list of ports to scan.

3. Scan Ports Function:
    - Iterate over each port in the list.
    - Create a socket based on the scan type (TCP or UDP).
    - Attempt to connect to the port or send a UDP probe.
    - If the connection attempt succeeds (result == 0), mark the port as "OPEN".
    - Optionally perform additional actions based on the scan configuration (service identification, banner grabbing, OS fingerprinting, vulnerability scanning).
    - Close the socket after each scan.
    - Error Handling: Catch and handle socket errors during port scanning.
4. Print Results: Display the status of each port after scanning is completed.

4. ChatGPT LLM Integration Algorithm

1. **Function Definitions:**
    - **"main(args)":**
        - Parse command-line arguments "args".
        - Extract "alert_file_location" and "apikey" from arguments.
        - Load JSON alert data from "alert_file_location".
        - Call "request_chatgpt_info" with the loaded data and "apikey".

    - **"debug(msg)":**
        - Check "debug_enabled" flag.
        - Print "msg" with current time if debugging is enabled.
        - Write "msg" to a log file.

    - **"collect(data)":**
        - Extract "port_service" and "choices" from "data".
        - Return "port_service" and "choices".

    - **"in_database(data, nmap_port_service)":**
        - Check if "port_service" exists in "data".
        - Return "True" if found, "False" otherwise.

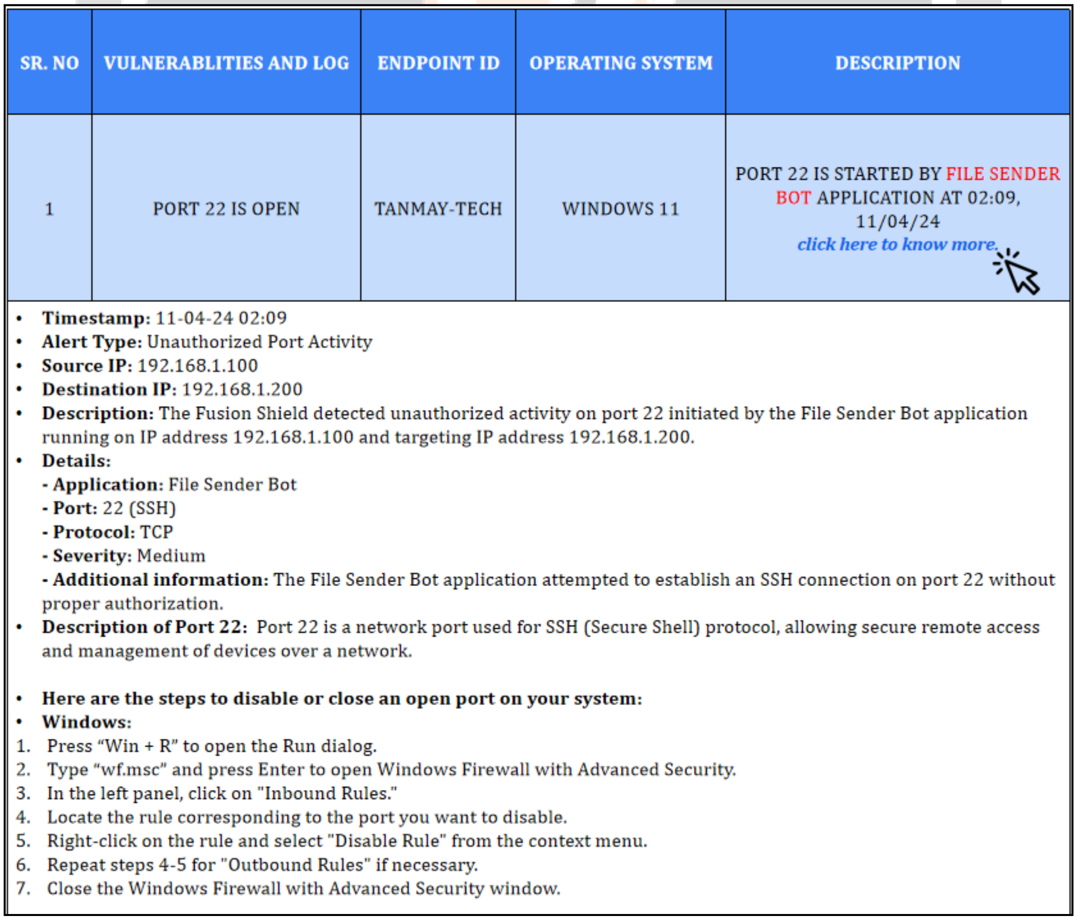
    - **"query_api(nmap_port_service, apikey)":**
        - Prepare headers with "apikey" for authentication.
        - Construct JSON data for API request containing "port_service".
        - Send a POST request to "https://api.openai.com/v1/chat/completions" with headers and JSON data.

- **"request_chatgpt_info(alert, apikey)":**
  - Initialize an empty "alert_output" dictionary.
  - Check if "port_service" exists in "alert['data']".
  - If exists, call "query_api" with "port_service" and "apikey".
  - Create an alert object based on the processed data and return "alert_output".

- **"send_event(msg, agent=None)":**
  - Construct a message based on "agent" details.
  - Log the message using "debug".
  - Send the message to a manager (not explicitly defined).

2. **Execution:**
   - Parse command-line arguments.
   - Initialize logging/debugging based on arguments.
   - Load JSON alert data from file.
   - Process the alert by calling "request_chatgpt_info" with the loaded data and API key.
   - Handle exceptions gracefully and log any errors encountered.

- This algorithm outlines the main flow of the program based on the provided functions and their interactions.
- Error handling and logging/debugging functionalities are integrated into the execution flow.
- Functions such as "query_api" and "send_event" involve interacting with external services and logging mechanisms, which are assumed to be implemented elsewhere.



SR. NO	VULNERABLITIES AND LOG	ENDPOINT ID	OPERATING SYSTEM	DESCRIPTION
1	PORT 22 IS OPEN	TANMAY-TECH	WINDOWS 11	PORT 22 IS STARTED BY FILE SENDER BOT APPLICATION AT 02:09, 11/04/24 *click here to know more.*

- **Timestamp:** 11-04-24 02:09
- **Alert Type:** Unauthorized Port Activity
- **Source IP:** 192.168.1.100
- **Destination IP:** 192.168.1.200
- **Description:** The Fusion Shield detected unauthorized activity on port 22 initiated by the File Sender Bot application running on IP address 192.168.1.100 and targeting IP address 192.168.1.200.
- **Details:**
  - **Application:** File Sender Bot
  - **Port:** 22 (SSH)
  - **Protocol:** TCP
  - **Severity:** Medium
  - **Additional information:** The File Sender Bot application attempted to establish an SSH connection on port 22 without proper authorization.
- **Description of Port 22:** Port 22 is a network port used for SSH (Secure Shell) protocol, allowing secure remote access and management of devices over a network.

- **Here are the steps to disable or close an open port on your system:**
- **Windows:**

1. Press "Win + R" to open the Run dialog.
2. Type "wf.msc" and press Enter to open Windows Firewall with Advanced Security.
3. In the left panel, click on "Inbound Rules."
4. Locate the rule corresponding to the port you want to disable.
5. Right-click on the rule and select "Disable Rule" from the context menu.
6. Repeat steps 4-5 for "Outbound Rules" if necessary.
7. Close the Windows Firewall with Advanced Security window.

Fig -2 Output Of Fusion Shield N-IDS

4. CONCLUSION

The development and implementation of the AI-Powered Network Intrusion Detection System (NIDS) mark a substantial leap forward in cybersecurity technology, harnessing the capabilities of artificial intelligence (AI) and machine learning (ML) to elevate network security and threat detection measures. Throughout the project lifecycle, several key objectives were accomplished, providing valuable insights into the intricacies of intrusion detection and response.

The AI-Powered NIDS successfully demonstrates the feasibility and efficacy of integrating AI technologies, such as machine learning algorithms and natural language processing (NLP), into conventional network intrusion detection systems. Leveraging both supervised and unsupervised learning techniques, the NIDS can perform real-time analysis of extensive network logs, identifying anomalies, classifying threats, and presenting actionable intelligence for security analysts.

A standout achievement of the project is the seamless integration of the OpenAI GPT-3 API, enriching intrusion alerts with detailed insights and recommended mitigation strategies in easily understandable formats. This integration significantly enhances the decision-making process for security analysts, facilitating prompt and effective responses to detected intrusions.

The modular design of the AI-Powered NIDS underscores scalability, maintainability, and flexibility, enabling straightforward integration of new functionalities, enhancements, and third-party integrations. Each module fulfills a distinct role within the system architecture, contributing to its robustness and reliability as an intrusion detection solution.

Throughout development, rigorous testing and validation procedures were conducted to ensure the functionality, performance, and security of the NIDS. Comprehensive test cases were executed to confirm compliance with specified requirements and pinpoint areas for refinement. Feedback from stakeholders and security analysts was carefully integrated to fine-tune system features and user interfaces.

In conclusion, the AI-Powered Network Intrusion Detection System signifies a pivotal milestone in advancing cybersecurity practices, equipping organizations with a sophisticated tool for proactive threat detection and incident response. The success of this project highlights the critical role of innovation and collaboration in addressing evolving cyber threats and safeguarding digital assets within our increasingly interconnected world.

6. REFERENCES

[1]"A Comprehensive Review of AI-Based Intrusion Detection System"
By M. S. Hossain, M. A. Razzak, and M. A. Al Mahmud
In: ScienceDirect, 2020

[2]"A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning"
By M. A. Bhuyan, S. K. Sikdar, and S. R. Ghose
In: MDPI, 2023

[3]"A Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges"
By M. Ahmed, M. A. Tahir, and N. A. Rahim
In: SpringerOpen, 2023

[4]"Intrusion Detection System using AI and Machine Learning Algorithm"
By M. D. Salam and M. A. Hossain
In: ResearchGate, 2023

[5]"Malware Detection and Prevention using Artificial Intelligence Techniques"
By M. J. H. Faruk, H. Shahriar, M. Valero, and F. Wu
In: Conference Paper, 2021

[6]"Ensemble Learning for Network Intrusion Detection"
By Y. Li, L. Zeng, L. Duan, and K. K. Leung
In: IEEE Transactions on Systems, Man, and Cybernetics: Part A: Systems and Humans, 2018, 48(10), 2395-2411

[7] "Deep Learning for Network Intrusion Detection"
By D. A. Arp, M. S. Portnoy, and S. E. Matthews
In: Proceedings of the 2014 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, 283-292

[8] "Anomaly-Based Network Intrusion Detection using Artificial Neural Networks"
By H. S. Park, J. S. Kim, and H. G. Kim
In: IEEE Transactions on Neural Networks and Learning Systems, 2011, 22(2), 242-250

[9] "Signature-based Intrusion Detection Systems: Challenges and Future Directions"
By P. Garcia-Tsaurán, J. López, A. Ureña, and R. J. Carbó
In: Journal of Network and Computer Applications, 2019, 128, 1-15

[10]"A Survey of Intrusion Detection Systems for Wireless Networks"
By D. D. González and J. C. A. López
In: IEEE Communications Surveys and Tutorials, 2010, 12(2), 296-347