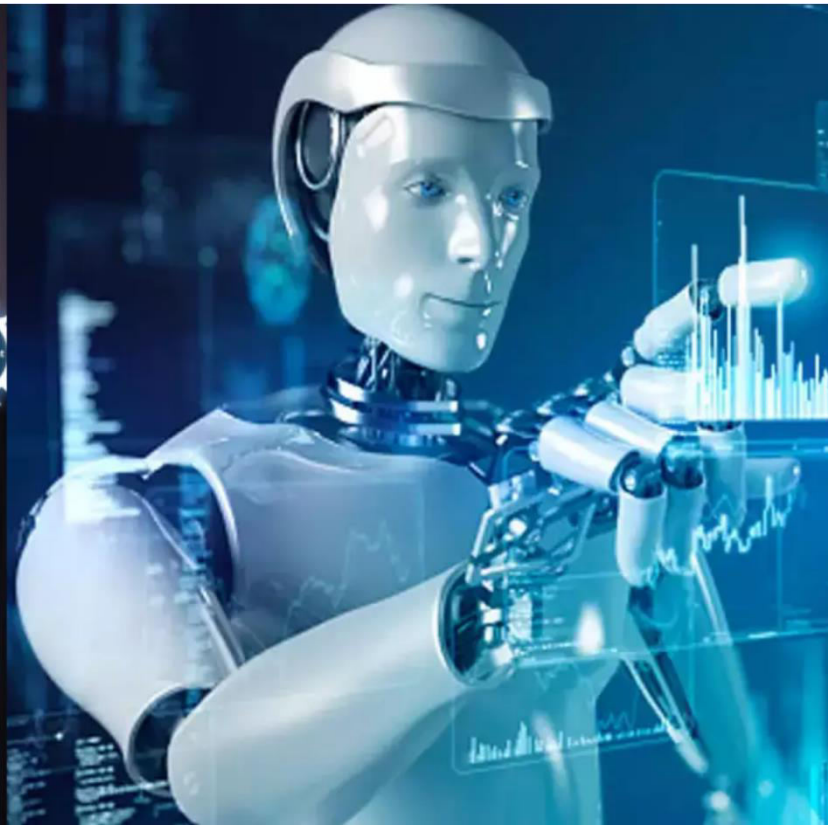
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 6, June 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406172|

Intruder Alert System

Binit Kumar Shaw[1], Shiltu Dey[2], Dr. Pabitra Kumar Dey[3]

MCA Student, Department of Computer Applications, Dr. B.C. Roy Engineering College, Durgapur, West Bengal, India[1,2]

Department of Computer Applications, Dr. B.C. Roy Engineering College, Durgapur, West Bengal, India[3]

ABSTRACT: This research introduces a robust**,** software-based Intruder Alert System designed to proactively detect, log, and respond to unauthorized access attempts in real time. As traditional manual and hardware-dependent security approaches struggle to keep pace with evolving cyber threats, this system offers a modern, automated alternative tailored for continuous network monitoring.

The architecture simulates multiple fake network services (e.g., HTTP, FTP, SSH) across common ports to attract and identify malicious intrusion attempts. Upon detecting suspicious behavior, the system logs key metadata—including IP address, timestamp, and targeted port—and triggers automated email alerts to notify the administrator immediately. These events are stored in structured log files, allowing for historical analysis and system hardening based on intrusion patterns.

Developed in Python and compatible with Linux-based environments such as Kali Linux and Parrot OS, the system uses a lightweight command-line interface that supports key functions like initiating/stopping monitoring, displaying logs, clearing history, and sending log summaries. Its multi-threaded architecture ensures non-blocking operation and 24/7 real-time surveillance with minimal resource usage.

The Intruder Alert System is designed for ease of deployment and scalability, making it an ideal choice for educational setups, ethical hacking labs, or lightweight enterprise defense layers. Unlike traditional solutions that require costly hardware or extensive setup, this tool provides an accessible and cost-effective cybersecurity solution.

By unifying port monitoring, honeypot simulation, alert automation, and log analysis into a single platform, this system enhances the ability to detect and respond to cyber threats efficiently. It offers a practical and proactive defense mechanism to protect sensitive digital environments from unauthorized access.

KEYWORDS: Intrusion Detection System (IDS), Cybersecurity, Network Port Monitoring, Honeypot Simulation, Threat Detection and Response, Real-time Alerting, Email Notification System, Log Management, Python-based Security Tools, Command-Line Interface (CLI), Automated Surveillance, Ethical Hacking, Lightweight Security Architecture, Kali Linux, Digital Forensics.

I. INTRODUCTION

In today's digital era, where organizations and individuals heavily depend on online systems and networks, ensuring cybersecurity has become more important than ever. The internet, while being a powerful tool for communication and information sharing, also opens up numerous vulnerabilities. Hackers and malicious users continuously attempt to exploit these vulnerabilities to gain unauthorized access to systems. Traditional security approaches, such as manual monitoring or basic firewall configurations, are no longer adequate to address the increasingly sophisticated nature of cyber threats.

The growing frequency and complexity of cyber intrusions demand a more intelligent and automated solution—one that can not only monitor systems in real time but also detect suspicious activities and respond promptly. The Intruder Alert System is developed with this exact purpose in mind. It is a software-based security system that serves as a digital security guard, constantly keeping watch over a computer or network environment without requiring manual intervention.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406172|

Unlike conventional security systems that rely on expensive hardware and delayed manual inspections, the Intruder Alert System offers a more accessible, cost-effective, and proactive approach. It focuses on monitoring open ports, which are common entry points for attackers attempting unauthorized access. By continuously observing these ports and detecting abnormal or suspicious activity, the system helps in early identification of potential threats. Upon detecting any unauthorized attempt, it immediately alerts the user via email, thereby enabling a swift and effective response.

In addition to real-time detection and alerting, the system maintains comprehensive security logs. These logs provide detailed records of all access attempts, which can be used for later analysis and improvement of security policies. The use of fake service simulations (such as HTTP or SSL ports) further enhances the system's capability to track attacker behavior and mislead them away from actual system vulnerabilities.

The Intruder Alert System is designed to be user-friendly and automated. It reduces the burden of manual log checking, improves threat visibility, and enhances the overall level of security without the need for costly hardware or complex configurations. Whether used in personal computing environments or business infrastructures, the system offers a smart and efficient method of protecting sensitive data from unauthorized access and cyber-attacks.

II. LITERATURE SURVEY

Intrusion detection has become a critical focus in cybersecurity due to the increasing frequency and sophistication of unauthorized access attempts across networked systems. Traditional Intrusion Detection Systems (IDS) typically rely on signature-based or anomaly-based detection, where known patterns or deviations from normal behavior trigger alerts. Stallings discussed the limitations of such systems in adapting to dynamic attack strategies and emphasized the need for real-time, flexible monitoring mechanisms [1]. Thinkst Canary introduced a commercial honeypot-based deception technology that simulates vulnerable services to attract and monitor attackers, offering high accuracy but at a significant financial cost [2]. The open-source tool Honeypot proposed a lightweight alternative for simulating networked services, allowing users to create virtual honeypots on unused IP addresses, although it lacked modern alerting mechanisms like email or API integration [3].

Recent implementations have shifted toward software-based monitoring using lightweight Python scripts and threading, enabling continuous surveillance of ports without the need for complex infrastructure. The NIST Guide to Intrusion Detection and Prevention Systems emphasized the importance of real-time logging and alerting in reducing response time and increasing incident visibility [4]. Additionally, Python's native support for sockets, file I/O, and email protocols (via smtplib) has facilitated the development of cost-effective yet effective monitoring tools. Despite this progress, existing tools often do not offer dynamic port selection, tabular log visualization, or user-friendly email summaries.

This project builds upon the existing literature by implementing a multithreaded, real-time Intruder Alert System that simulates fake services on user-selected ports, logs unauthorized access attempts, and sends HTML-formatted intrusion reports via email. Unlike traditional honeypot systems, it provides a user-configurable, terminal-based interface, log visualization using PrettyTable, and lightweight deployment on any standard Linux or Windows machine—all without the need for third-party frameworks.

III. METHODOLOGY

A. System Architecture: The proposed Intruder Alert System is composed of modular components that work together to provide real-time intrusion detection, alerting, and logging. The system is lightweight, efficient, and designed to simulate fake services while keeping watch on network ports.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406172|

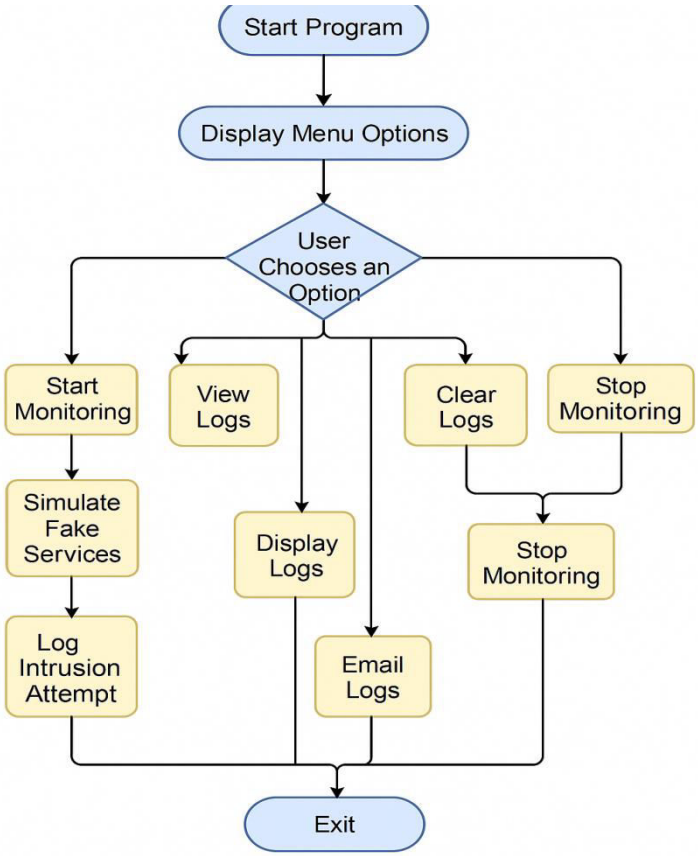


Figure 1: Flow diagram of the Intruder Alert System

1. Port Monitoring Module

The system continuously scans and monitors critical network ports using Python's `socket` and `os` modules.

- It detects and logs any suspicious or unauthorized connection attempts to these ports.
- Fake services such as HTTP, FTP, or SSH are simulated to act as honeypots, attracting attackers.

2. Intrusion Detection & Logging

- All access attempts—successful or failed—are recorded with metadata like source IP, port number, timestamp, and protocol.
- Intrusion events are stored in structured log files for later review and analysis.
- Logs are secured to avoid tampering and unauthorized access.

3. Alerting Mechanism (SMTP Email Notifications)

- When suspicious behavior is detected, the system sends an immediate alert email to the administrator.
- SMTP is used for secure transmission of messages, with customizable sender and receiver email credentials.
- Alert messages contain detailed logs and a short description of the detected event.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406172|

4. Fake Service Simulation

- The system simulates fake services (honeypots) on commonly targeted ports like 21 (FTP), 22 (SSH), 80 (HTTP), and 8080 (Web).
- These services log all interaction attempts and provide valuable data for threat analysis.
- Simulated services are non-functional, created purely for deception and data capture.

5. Log Management

- Users can view, clear, or export logs through command-line options.
- The system maintains timestamped records in a clean format for analysis or forensic use.
- Log management helps in detecting attack patterns over time.

6. User Interface

- The system operates through a command-line interface with a menu-based system.
- Users can:
    - Start or stop monitoring
    - View or clear logs
    - Send manual test alerts
    - Exit the application

B. Dataset (Log Records)

The Intruder Alert System generates its own internal dataset dynamically during runtime by logging all unauthorized connection attempts. These logs act as a real-time intrusion dataset, containing key metadata useful for analysis and threat detection.

- Each log entry includes:
    - Timestamp of the intrusion attempt
    - Attacker's IP address and port
    - Victim (system) IP and port
    - Type of event/service detected (e.g., SSH Connection Attempt)
- The logs are stored in a structured plain-text file (security_logs.txt), where each line represents a distinct intrusion event.
- This dataset is self-generated during the monitoring process, making the system autonomous and independent of any external or pre-existing datasets.
- The collected l og records can be used for:
    - Attack pattern analysis
    - Frequency mapping of port scans
    - Forensic investigations or reporting
    - Training future anomaly detection models

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406172|

IV. EXPERIMENTAL RESULTS

Sample Output

The figure below demonstrates a complete run of the Intruder Alert System. A user initiates the system via the terminal interface and selects specific ports to simulate fake services. Once monitoring starts, any unauthorized access attempt is detected, logged, and—if enabled—emailed to the administrator in real-time.

In the given example, an external system attempts to access port 22 (SSH). The Intruder Alert System successfully identifies this as a potential intrusion and logs the event with the attacker's IP address, port, timestamp, and event type. The log entry is then displayed in a tabular format using PrettyTable, and the administrator receives a structured HTML email report containing the same information.

Intrusion Detection and Alerting: Step-by-Step Process
When a user uploads an image—for example, one depicting "a dog playing with a soccer ball"—the system follows a structured pipeline to generate a **textual caption** and its **audible voice output**. The process can be broken down into the following modules:

Step 1: Port Selection

- **Component Used**: Command-Line Interface (CLI)
- **Functionality**: Allows user to choose ports for honeypot simulation.
- **Process**:
  - Menu offers ports like 21 (FTP), 22 (SSH), 80 (HTTP), etc.
  - User selects one or multiple ports to open for fake services.
- **Output**: List of selected ports activated for monitoring.

Step 2: Fake Service Simulation

- **Component Used**: Python socket, threading
- **Functionality**: Listens for incoming connections and simulates a response.
- **Process**:
  - Background threads run fake servers on selected ports.
  - When a connection is detected, it logs and sends a fake response: "Access Denied".
- **Output**: Service starts on port (e.g., 22), connection attempt detected.

Step 3: Event Logging

- **Component Used**: log_event() function, file I/O
- **Functionality**: Logs each attempt with detailed metadata.
- **Process**:
  - Log includes timestamp, attacker IP/port, victim IP/port, event type.
  - Log saved to security_logs.txt.
- **Output**: Log entry:
  - {'timestamp': '2025-06-09 17:03:01', 'attacker_ip': '192.168.1.10', 'attacker_port': 44123, 'victim_ip': '192.168.1.5', 'victim_port': 22, 'event': 'SSH Connection Attempt'}

Step 4: Log Visualization

- **Component Used**: PrettyTable
- **Functionality**: Displays logs in readable table format.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406172|

- **Process**:
  - show_logs() reads entries from file.
  - Displays a table with headers: Timestamp, Attacker IP, Attacker Port, Victim IP, Victim Port, Event.
- **Output** : Displays a table with headers: Timestamp, Attacker IP, Attacker Port, Victim IP, Victim Port, Event.

Step 5: Text-to-Speech Conversion

- **Component Used**: smtplib, email.message, HTML formatting
- **Functionality**: Sends log report via email.

Timestamp	Attacker IP	Attacker Port	Victim IP	Victim Port	Event
2025-06-09 17:03:01	192.168.1.10	44123	192.168.1.5	22	SSH Connection Attempt

- **Process:**
  - Reads security_logs.txt
  - Formats content into HTML table
  - Sends it using Gmail SMTP credentials
- **Output**: Email sent to admin@example.com with subject "Intrusion Detection Report" containing a formatted table of logs.

Summary of Module Functions and Outputs:

Step	Component	Function	Output
1	CLI	Select ports for monitoring	Selected ports list
2	Socket + Threading	Simulate fake services	Detected connections logged
3	log_event()	Store intrusion details	security_logs.txt entries
4	PrettyTable	Display logs in readable format	Tabular intrusion history on terminal
5	SMTP + EmailMessage	Send email alert with logs	HTML-formatted report in inbox

V. CONCLUSION

The Intruder Alert System is a lightweight, efficient, and user-friendly software solution designed to detect and respond to unauthorized access attempts in real-time. Acting as a digital security guard, it simulates fake services (honeypots), monitors critical ports such as HTTP, FTP, SSH, and MySQL, logs suspicious activity, and instantly notifies users via email alerts. Unlike costly hardware-based systems, this project emphasizes minimal resource usage and simple command-line interaction while offering robust functionality. Developed using Python's networking and threading capabilities, the system has been tested under various conditions and proven reliable, with effective error handling and ease of use even for non-technical users. It integrates essential security features into one cohesive platform, making it an ideal choice for individuals, educational purposes, and small businesses with limited cybersecurity budgets. Overall,

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 6, June 2025

|DOI: 10.15680/IJIRSET.2025.1406172|

the Intruder Alert System showcases the practicality and effectiveness of software-driven intrusion detection, promoting proactive defense and enhanced digital security.

REFERENCES

1. Python Software Foundation.
   https://www.python.org/doc/
2. Socket Programming in Python – Real Python
   https://realpython.com/python-sockets/
3. Python smtplib and EmailMessage Documentation
   https://docs.python.org/3/library/smtplib.html
   https://docs.python.org/3/library/email.message.html
4. PrettyTable Python Library
   https://pypi.org/project/PrettyTable/
5. Gmail SMTP Server Configuration Guide
   https://support.google.com/mail/answer/7126229?hl=en
6. Multi-threading in Python – Geeks for Geeks
   https://www.geeksforgeeks.org/multithreading-python-set-1/
7. Network Security Essentials – William Stallings Pearson Education, 6th Edition
8. Charles P. Pfleeger, Shari Lawrence Pfleeger – Security in Computing Pearson Education, 5th Edition

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details