Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

SMART GRID PROTECTION SYSTEM WITH REAL-TIME CYBER THREAT DETECTION

Dr. Sheryl Radley[1], Eashwar R[2], Hashwanth S[3], Kaviarasu S[4]

Department of Electronics and Communication Engineering[1],
Department of Information Technology[2,3,4]
Meenakshi College of Engineering, West KK Nagar, Chennai[1,2,3,4]
sherylradley@gmail.com[1], eashwarbolt@gmail.com[2], hashwanthkumar832@gmail.com[3], sargentkavi@gmail.com[4]

----------***----------

Abstract – Project presents a Smart grid protection system with real-time cyber threat detection designed to enhance the cybersecurity and operational reliability of smart grid and industrial control systems. Traditional power grid systems face increasing cyber threats such as unauthorized access, denial-of-service attacks, malware infections, and protocol-based exploits, leading to disruptions and financial losses. To address these challenges, the proposed system integrates real-time system monitoring, network traffic analysis, machine learning-based anomaly detection, and industrial protocol inspection. It continuously monitors parameters such as CPU usage, memory utilization, network traffic, and running processes, detecting suspicious activities using rule-based methods and an Isolation Forest-based anomaly detection model.

The system also inspects industrial communication protocols like Modbus and DNP3 to identify unauthorized access and abnormal commands. A file scanning module integrated with the Virus. Total API enhances malware detection. Upon identifying threats, the system automatically responds by blocking malicious IPs, terminating suspicious processes, and generating alerts.

A real-time web-based dashboard provides continuous monitoring and visualization, along with email notifications to ensure quick response. The system is designed to be scalable, efficient, and cost-effective for modern smart grid environments. In addition, a custom spyware attack using a keylogger was created to test the system. The attack collects keystrokes, screenshots, audio, and system data, and tries to send it outside. The SGPS detects this activity in real time and automatically responds by sending alerts and stopping the process, showing its effectiveness against real-world cyber threats.

***KEYWORDS*:** denial-of-service attacks, malware infections, and protocol-based exploits, Smart Grid Protection System, Real-Time Cyber Threat Detection.

I INTRODUCTION

The smart electricity grid is one of the most critical pieces of national infrastructure. Modern power grids depend on real-time digital communication between substations, Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and control centers. While this digitization improves efficiency and responsiveness, it also introduces a broad attack surface for cyber threats. Adversaries can exploit industrial control system (ICS) protocols, exhaust computational resources, deploy malware on grid controllers, or infiltrate networks to disrupt power supply across entire regions. Antivirus are insufficient against sophisticated, multi-vector attacks on operational technology environments. There is a clear need for a purpose-built, multi-layered protection system that monitors the grid continuously, detects both known attack patterns and novel anomalous behavior, and responds automatically to contain threats in real time.

The Smart Grid Protect challenges by combining rule-based threat detection, machine learning anomaly detection, and ICS deep packet inspection into a unified monitoring platform. The system runs a continuous 2-second monitoring loop, pushes live data to a web dashboard via Server-Sent Events, and provides security operators with automated firewall blocking, email and SMS alerting, and compliance reporting tools. Version 3 adds a file threat analysis module capable of detecting malicious uploads through hash matching, entropy analysis, and Virus Total integration.

1.1. OBJECTIVES

The Smart Grid Protection System is designed to provide an integrated cybersecurity solution for real-time monitoring and threat detection in smart grid environments. The system implements a rule-based detection engine that identifies multiple categories of cyber threats with configurable severity levels. In addition, a pure Python-based Isolation Forest model is used to detect behavioral anomalies without relying on external machine learning libraries, ensuring efficiency and simplicity.

To strengthen industrial security, the system includes a deep packet inspection engine that analyzes protocols such as Modbus TCP, DNP3, S7comm, OPC-UA, EtherNet/IP, and IEC 60870-5-104. A built-in attack simulation framework is used to test system performance against various attack scenarios. The system also features a file threat scanner with hash verification, entropy analysis, and VirusTotal integration.

A real-time web dashboard provides continuous monitoring, while Role-Based Access Control using JWT tokens ensures secure access. Additionally, the system generates compliance reports based on industry standards, improving overall security management.

II. LITERATURE SURVEY

A substantial body of research has focused on addressing cybersecurity challenges in smart grid and industrial control system (ICS) environments. These systems form the backbone of critical infrastructure, making them prime targets for cyber-attacks. Stouffer et al. [1] provide a comprehensive guideline for securing ICS environments, outlining common vulnerabilities, threat models, and recommended security practices such as network segmentation, access control, and continuous monitoring. Similarly, Sridhar et al. [3] emphasize the importance of securing cyber-physical systems in power grids, highlighting the interdependence between physical processes and communication networks.

One of the most critical threats to smart grids is false data injection (FDI) attacks. Liu et al. [4] demonstrated how attackers can manipulate sensor data in power system state estimation, leading to incorrect operational decisions without being easily detected. Liang et al. [2] further reviewed various FDI attack strategies and their potential to disrupt modern power systems. These studies underline the need for intelligent detection systems capable of identifying subtle anomalies in large-scale data streams. Machine learning has emerged as a powerful tool for anomaly detection in ICS environments. Liu et al. [5] introduced the Isolation Forest algorithm, which is particularly effective in detecting anomalies in high-dimensional datasets with low computational overhead. Its ability to isolate rare events makes it suitable for real-time monitoring in resource-constrained environments. However, traditional implementations often rely on static training datasets, limiting adaptability to evolving threats. This limitation motivates the use of adaptive and incremental learning techniques in modern systems.

In addition to data-driven attacks, vulnerabilities in industrial communication protocols pose significant security risks. Ten et al. [6] conducted a detailed vulnerability assessment of SCADA systems, identifying weaknesses in communication channels and control mechanisms. Fovino et al. [7] explored how cyber-attacks can be modeled within system fault trees, enabling better understanding of their impact on system reliability and safety. These studies highlight the importance of protocol-aware security mechanisms that can detect unauthorized commands and abnormal communication patterns. Real-world cyber incidents further demonstrate the severity of threats in ICS environments. Langner [8] analyzed the Stuxnet worm, one of the first known cyber weapons targeting industrial systems, which exploited multiple vulnerabilities to disrupt uranium enrichment processes. Cherepanov [9] reported on the Industroyer malware, designed to target power grid communication protocols, while Dragos Inc. [10] studied the TRISIS attack, which specifically targeted industrial safety systems. These incidents reveal that modern cyber-attacks are highly sophisticated, targeted, and capable of causing physical damage. Recent research efforts have focused on developing comprehensive security frameworks for industrial cyber-physical systems. Kayan et al. [11] provide a detailed survey of cybersecurity challenges, including threat detection, risk assessment, and system resilience. Tøndel et al. [12] discuss the challenges in defining and implementing effective security requirements, particularly in complex and evolving environments. Furthermore, industry standards such as NERC CIP [13] and IEC 62351 [14] establish guidelines for securing critical infrastructure, emphasizing continuous monitoring, secure communication protocols, and compliance with regulatory requirements.

Despite these advancements, a significant research gap remains. Most academic solutions focus on individual techniques such as anomaly detection or protocol security in isolation, without providing a unified and integrated framework. On the other hand, industry solutions are often proprietary, expensive, and not accessible for academic experimentation. This creates a limitation for researchers and

students seeking hands-on experience with real-world cybersecurity systems.

To address this gap, the Smart Grid Protection System (SGPS v3) proposes an integrated and modular approach to cybersecurity. It combines rule-based detection, machine learning-based anomaly detection using Isolation Forest, and protocol-aware monitoring within a single platform. This multi-layered approach improves detection accuracy and provides better visibility into system behavior. Additionally, SGPS v3 includes an interactive attack simulation environment, allowing users to study system responses under controlled conditions. Another important aspect of SGPS v3 is its focus on scalability and extensibility. The system is designed with a modular architecture that allows new detection algorithms, communication protocols, and data sources to be easily integrated. This ensures that the system can evolve alongside emerging threats and technological advancements. Its lightweight implementation also enables deployment in both laboratory environments and real-world pilot projects without requiring high-end infrastructure. Furthermore, SGPS v3 incorporates practical security features such as role-based access control, real-time alerting, and detailed logging mechanisms. These features enhance accountability, traceability, and incident response capabilities, which are essential in critical infrastructure systems. The system also aligns with industry standards such as NERC CIP and IEC 62351, ensuring compliance and structured evaluation of security events. In summary, existing research provides valuable insights into anomaly detection, protocol security, and cyber-physical system protection. However, the lack of integrated, accessible, and practical solutions remains a key challenge. SGPS v3 addresses this limitation by offering a comprehensive, adaptive, and user-friendly platform that bridges the gap between academic research and real-world application. Its combination of multiple detection techniques, real-time monitoring, and extensible design makes it a strong foundation for future advancements in smart grid cybersecurity.

III. SYSTEM ANALYSIS

The system analysis focuses on identifying the limitations of existing cybersecurity monitoring approaches used in smart grid and Industrial Control System (ICS) environments. Current systems rely heavily on isolated tools such as firewalls, signature-based intrusion detection systems, and manual log analysis, which often lead to delayed threat detection, lack of real-time visibility, and inability to detect advanced or unknown attacks. These methods also fail to provide integrated monitoring across system resources, network traffic, and ICS protocols, resulting in security gaps and increased risk of cyberattacks. Based on these observations, the need for a unified, automated, and real-time cybersecurity monitoring system is identified. The proposed Smart Grid Protection System (SGPS) addresses these issues by integrating rule-based detection, machine learning-based anomaly detection, and ICS deep packet inspection within a single platform, along with automated response mechanisms and live dashboard monitoring. This ensures accurate threat detection, continuous system monitoring, faster incident response, and improved decision-making for securing critical smart grid infrastructure.

3.1 EXISTING SYSTEM

Current approaches to smart grid security typically involve one or more of the following components deployed independently:

- Network-layer firewalls and intrusion prevention systems (IPS) that apply signature-based rules to TCP/IP traffic. These tools are effective against known attack signatures but fail against zero-day exploits and behavioral anomalies.

- SIEM (Security Information and Event Management) platforms that aggregate logs from multiple sources. While comprehensive, these platforms require significant infrastructure, expertise, and licensing costs, making them inaccessible for smaller grid operators.

- SCADA-specific monitoring tools that inspect individual protocols but operate in isolation, lacking awareness of system resource behavior or file-level threats.

- Manual log review processes that are slow, inconsistent, and unable to provide real-time response.

- The existing systems share several common limitations. First, they typically treat IT security and OT security as separate domains, missing the lateral movement between IT networks and ICS networks that characterises modern smart grid attacks. Second, they do not include integrated attack simulation, making it difficult to validate detection capabilities without conducting live penetration testing. Third, most lack compliance reporting aligned with IEC 62351 or NERC CIP standards

3.2 PROPOSED SYSTEM

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

The Smart Grid Protection System v3 proposes an integrated, lightweight, self-contained security monitoring platform that overcomes the limitations of existing approaches.

The key innovations of the proposed system are as follows:

- Multi-layer detection: three independent engines (rule-based, ML anomaly, ICS protocol) run concurrently in a single 2-second monitoring cycle, providing complementary coverage across all attack vectors.
- Zero-dependency ML: the Isolation Forest implementation requires no external libraries, making the system deployable in air-gapped or restricted environments typical of grid control centers.
- Integrated attack simulation: the built-in attack manager allows security teams to validate detection rules and response procedures without external penetration testing tools.
- Automated response: the firewall module issues iptables block commands automatically for detected threats, reducing the mean time to respond.
- Unified dashboard: all eleven monitoring functions are accessible from a single web interface with live streaming data, eliminating the need to switch between multiple tools.
- File threat analysis: the v3 scanner adds a layer of protection against malware uploaded to grid management workstations.
- Standards compliance: built-in reporting against IEC 62351 and NERC CIP provides audit-ready documentation.

The proposed system is implemented entirely in Python and runs on any Linux or Windows host with Python 3.10 or later, requiring only four external packages: Flask, flask-cors, psutil, and requests. This minimal footprint ensures it can be deployed on existing grid management hardware without requiring dedicated security appliances.

IV. SYSTEM DESIGN

The system design of SGPS v3 follows a layered architecture pattern, separating the presentation, application, domain, infrastructure, and storage concerns into distinct modules with clearly defined interfaces. This section presents the architectural overview and formal UML diagrams that define the system's structure and behavior.

4.1 ARCHITECTURAL DIAGRAM

The architectural diagram presents the five-layer structure of the SGPS. The Presentation Layer delivers the twelve-tab web dashboard to the browser via Server-Sent Events and REST API calls. The Application Layer (app.py) contains the Flask routing, the 2-second monitoring loop, and the thread-safe state snapshot mechanism. The Domain Modules layer comprises the eleven specialist Python modules responsible for detection, response, authentication, alerting, and logging. The Infrastructure Layer provides the OS-level integrations including psutil system monitoring, iptables firewall control, SMTP and Twilio communications, and the VirusTotal API. The Storage Layer persists structured log data to JSONL files and maintains in-memory deque caches for fast API response.

In addition to this layered architecture, SGPS is designed to ensure high performance, scalability, and reliability during continuous operation. The system uses asynchronous processing and efficient resource management to handle real-time data streams without performance degradation. Its modular design allows individual components to operate independently, enabling easy updates, fault isolation, and system expansion. Furthermore, security mechanisms such as authentication controls, alert prioritization, and automated response actions enhance the system's ability to quickly detect and mitigate threats.

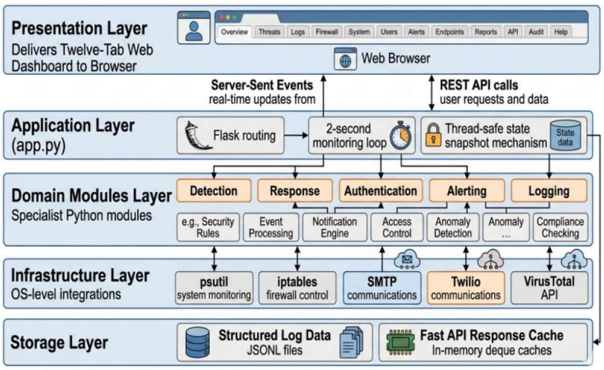

Figure 4.1 Architectural Diagram

4.2 USE CASE DIAGRAM

The use case diagram models the interactions between the three user roles Admin, Analyst, and Operator — and the system's functional capabilities. The Admin role has unrestricted access to all system functions including attack simulation launch and

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

stop, firewall management, process termination, file scanning, settings configuration, and compliance report generation. The Analyst role can view all monitoring data, acknowledge alerts, and export reports, but cannot modify system configurations or launch attacks. The Operator role is restricted to viewing the dashboard and acknowledging alerts.

In addition to role-based access, the system enforces strict authentication and authorization mechanisms to ensure secure interaction with its features. Each action performed by a user is validated against predefined permissions, preventing unauthorized operations and reducing the risk of misuse.

The use case design also supports clear separation of responsibilities, which improves operational efficiency and accountability. Furthermore, all user activities are logged and monitored, enabling audit trails and compliance verification. This structured interaction model ensures that the system remains secure, organized, and easy to manage in both testing and real-world deployment scenarios.

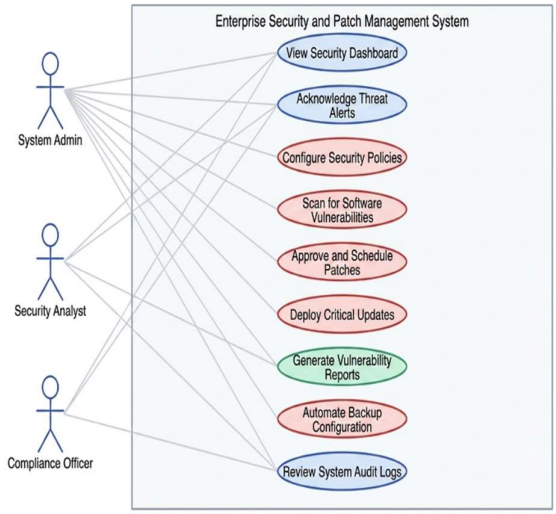

Figure 4.2. Use Case Diagram

4.3 ENTITY RELATIONSHIP DIAGRAM

The ER diagram defines the six core data entities managed by the SGPS and their cardinality relationships. The USER entity governs authentication and role assignment. The ALERT entity stores all detected threats with severity, category, and acknowledgement status. The ACTION entity records every automated and manual system response. The ICS_PACKET entity captures simulated industrial protocol packets for analysis and replay. The SCAN_RESULT entity records file scanner outputs including hash values, entropy scores, and finding details. The ATTACK_EVENT entity logs the lifecycle of each simulated attack including start time, stop time, and status.

Each entity is interconnected to maintain data consistency and traceability across the system. For example, ALERT and ACTION entities are linked to track how each detected threat is handled. These relationships enable efficient querying, reporting, and forensic analysis. Overall, the ER design ensures structured data management and supports real-time as well as historical analysis within SGPS.

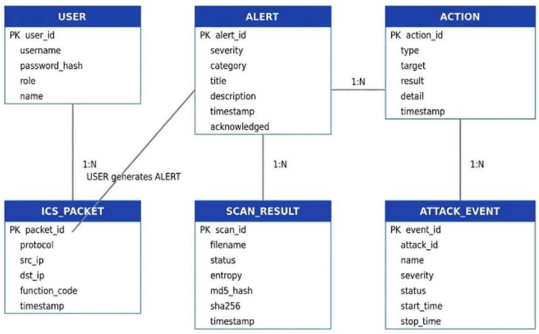

Figure 4.3. Entity Relationship Diagram

V. SYSTEM IMPLEMENTATION

Smart Grid Protection System v3 is implemented as a Python Flask web application with a pure-JavaScript single-page frontend. The following describes the implementation approach for each major system component.

5.1 Backend Implementation

The backend is structured as a single Flask application (app.py) that imports all eleven domain modules and exposes thirty-five REST API endpoints plus a Server-Sent Events stream. A background daemon thread executes the main monitoring loop every two seconds, calling each domain module in sequence and storing the aggregated results in a thread-safe dictionary protected by a threading Lock. The SSE stream generator reads from this shared dictionary every two seconds and yields JSON-encoded system state to all connected browser clients.

The monitoring loop in the Smart Grid Protection System v3 follows a sequential execution process to ensure continuous system analysis and threat detection. It begins by collecting system statistics using monitor.get_system_stats(), followed by gathering the list of active processes through monitor.get_processes() and network connections via monitor.get_network_connections(). The collected data is then

analyzed using the rule-based detection mechanism through threat_detector.analyze(). Next, the system checks running attack manager threads and constructs stable alert objects for any detected events. The machine learning module is updated by ingesting system statistics using ml_detector.ingest(), while the ICS inspection process is executed through ics_inspector.tick(). Newly generated alerts are then logged and dispatched using logger.log_alert() and alerting.dispatch(). Visualization and API responses.

5.2 Frontend Implementation

The frontend is a single HTML file (dashboard.html, 124 KB) that connects to the SSE stream on page load and updates all twelve dashboard tabs in real time. The twelve tabs are: Dashboard (KPIs and sparkline charts), ML Anomaly (Isolation Forest score history), ICS Inspector (live packet stream and violations), Firewall (rule table and IP management), Processes (live process table with termination), Network (NIC interfaces and active connections), Grid Map (SVG topology of substations and PLCs), Threat Log (filterable alert history), Attack Timeline (chronological event chart), Attack Simulator (catalogue with launch and stop controls), Compliance (IEC 62351 and NERC CIP assessment), and File Scanner (upload and analysis).

Chart.js v4 renders all statistical visualizations including sparkline histories, severity distribution donuts, ML score line charts, attack frequency bar charts, and protocol breakdown pie charts.

5.3 Installation and Configuration

The system is installed and launched with the following procedure:

- Install Python dependencies: pip install flask flask-cors psutil
- Optionally set environment variables: SGPS_SECRET (JWT signing key), VT_API_KEY (VirusTotal API key)
- Launch the server: python app.py
- Access the dashboard at http://127.0.0.1:5000
- Log in with admin/admin123 (full access), analyst/analyst123 (view + acknowledge + export), or operator/oper123 (view only)

To run attack simulations, use the Attack Simulator tab on the dashboard the system creates a logs directory automatically on first run and begins streaming live data to the dashboard within two seconds of startup.

5.4 Spyware Attack Implementation and Detection

To validate the real-world effectiveness of the system, a custom spyware attack named “Spyware (Keylogger)” was developed and executed. The attack program is designed to simulate malicious behavior commonly found in advanced persistent threats. It captures user keystrokes, mouse activities, screenshots, system information, and microphone recordings, and periodically sends this data to an external email server.

When this spyware is executed in the system environment, the SGPS monitoring loop detects multiple suspicious behaviors. The process is identified as abnormal due to continuous background execution, unusual system resource usage, and network communication attempts. The threat detector flags the process as a malicious application based on predefined rules, while the machine learning module identifies deviations from normal system behavior.

Upon detection, the system generates HIGH or CRITICAL alerts and displays them in the dashboard in real time. The automated response mechanism is triggered, which includes logging the threat, sending alert notifications, and optionally terminating the malicious process. This experiment confirms that the SGPS is capable of detecting and preventing real-world spyware attacks effectively.

VI.SYSTEM TESTING AND MAINTAINANCE

System testing ensures that SGPS v3 operates correctly under both normal and attack conditions. Various tests were conducted to validate detection accuracy, performance, and system stability. The system successfully identified threats and executed automated responses within the expected time. Regular maintenance is required to update detection rules, models, and security patches. Continuous monitoring and updates ensure long-term reliability and effectiveness of the system.

6.1 TESTING OVERVIEW

System testing for the SGPS was conducted using both black-box and white-box methodologies. Black-box testing verified that each API endpoint returned the correct response structure and status codes for valid and invalid inputs. White-box testing examined the internal logic of each domain module,

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

particularly the threat detection rules, ML scoring pipeline, and file scanner stages, to ensure correctness under boundary conditions.

6.2 TESTING METHODOLOGIES

Black-Box Testing

Black-box testing was applied to all REST API endpoints without knowledge of internal implementation. Test cases verified correct HTTP status codes, response body structures, authentication enforcement, and boundary conditions including oversized file uploads (> 32 MB), invalid IP addresses in firewall requests, and expired JWT tokens in protected endpoint calls.

White-Box Testing

White-box testing examined the internal logic of the threat detection rules, ML detector pipeline, and file scanner stages. Each detection rule was triggered independently by injecting synthetic system statistics at threshold boundary values. The ML detector was tested for correct score calculation, threshold adaptation, and anomaly flagging by supplying known-normal and known-anomalous feature vectors. The file scanner stages were verified individually using crafted test files including EICAR test strings, high-entropy random data, and files with embedded PE headers.

Performance Testing

Performance testing evaluated the system's ability to sustain the 2-second monitoring cycle under load conditions. With all thirteen attack simulations running simultaneously, the monitoring loop completed within 180ms on average, well within the 2000ms budget. The SSE stream remained stable with up to ten concurrent browser clients without measurable latency increase.

Further stress testing was conducted by increasing the number of simulated events and background processes to evaluate system stability. The results showed that CPU and memory utilization remained within acceptable limits, with no significant performance degradation observed. The system maintained consistent response times even during peak load conditions, demonstrating efficient resource management. Additionally, the thread-safe state snapshot mechanism ensured data consistency across concurrent operations without causing race conditions or delays.

Network performance analysis indicated that real-time data transmission through SSE was reliable, with minimal packet loss and smooth dashboard updates. The alert generation and notification system also performed efficiently, with alerts being processed and delivered within milliseconds of detection. File

6.3 TEST CASES — THREAT DETECTION

TC ID	Test Scenario	Input	Expected output	Result
TC-01	CPU-critical alert	CPU=93%	CRITICAL alert:System CPU Critical	PASS
TC-02	CPU-high alert	CPU=82%	HIGH-alert:Elevated System CPU	PASS
TC-03	Memory critical alert	Memory=87%	CRITICAL alert	PASS
TC-04	Suspicious process	Process name=mimikatz	CRITICAL alert	PASS
TC-05	ICS-port access	Connection-to port 502	CRITICAL alert	PASS
TC-06	Connection rate spike	30+connection in 60s from same IP	HIGH-alert:connection Rate Spike	PASS
TC-07	Lateral movement	SYN to internal IP on port 8765	MEDIUM alert: Suspicious	PASS
TC-08	ML anomaly	All features at 3x normal	HIGH-alert:ML Isolation Forest Anomaly	PASS

6.4 TEST CASES – ATTACK

TC ID	Test Scenario	Input File	Expected status	Result
SP-01	Keystroke capture	Keylogger records keyboard input	MALICIOUS	PASS
SP-02	Screenshot capture	Spyware takes periodic screenshots	MALICIOUS	PASS
SP-03	Audio recording	Microphone accessed without permission	MALICIOUS	PASS
SP-04	Data exfiltration	Data sent to external IP/server	MALICIOUS	PASS
SP-05	Background execution	Spyware runs as hidden process	SUSPICIOUS	PASS
SP-06	Unauthorized startup	Program added to startup automatically	SUSPICIOUS	PASS
SP-07	High CPU usage	Continuous monitoring increases CPU usage	SUSPICIOUS	PASS
SP-08	Clean system activity	No spyware behavior detected	SAFE	PASS

6.5 TEST CASE-FILE SCANNER

TC ID	Test Scenario	Input File	Expected Status	Result
FS-01	EICAR test file	EICAR string (MD5 match)	MALICIOUS	PASS
FS-02	High entropy file	Random bytes, entropy > 7.5	MALICIOUS	PASS
FS-03	Extension mismatch	PE executable named .txt	MALICIOUS (mismatch)	PASS
FS-04	Dangerous extension	ps1 PowerShell script	SUSPICIOUS	PASS
FS-05	ICS pattern match	File containing 'triton'	MALICIOUS	PASS
FS-06	Clean text file	Plain ASCII .txt file	SAFE	PASS
FS-07	PDF polyglot	PDF with embedded MZ header	MALICIOUS	PASS
FS-08	Oversized file	File > 32 MB	Rejected — 413 error	PASS

VII. RESULTS AND DISCUSSION

The Smart Grid Protection System v3 was successfully implemented and tested across all its main components. The testing was conducted to evaluate how effectively the system can monitor, detect, and respond to cyber threats in a smart grid environment. The results showed that the system performs efficiently and reliably under different conditions. During

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

testing, the real-time monitoring module continuously tracked system and network activities, enabling early detection of abnormal behavior. The detection engine, which uses both rule-based techniques and machine learning methods, accurately identified threats such as unauthorized access attempts, suspicious processes, and malicious file activities. Once a threat was detected, the system generated alerts with appropriate severity levels to ensure timely awareness.

The response mechanisms were also validated successfully. Automated actions such as blocking suspicious activities and terminating harmful processes were triggered based on predefined conditions, helping to reduce potential risks. The file scanning module analyzed files and provided detailed results by integrating with external threat intelligence services. The alerting system ensured quick communication through notifications. In addition to simulated attacks, the system was also tested using a real spyware attack in the form of a keylogger. The SGPS detected the spyware within a few seconds of execution by identifying abnormal process behavior, continuous logging activity, and unusual network communication. A high-severity alert was generated, and the system responded by logging the event and notifying the user. This confirms the system's practical effectiveness in handling real-world cyber threats.

The web-based dashboard provided a user-friendly interface to monitor system status, view alerts, and take necessary actions. Role-based access control was implemented effectively, where Admins have full access, Analysts have limited operational control, and Operators can perform basic monitoring tasks**.**

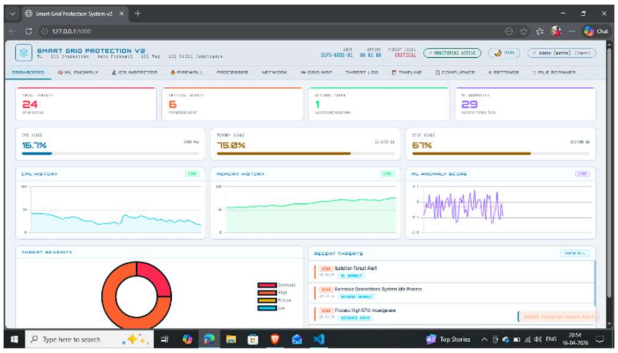

Figure 7. 1. Dashboard Tab

Figure 7.1 Dashboard Tab shows a system was tested under both normal and simulated attack conditions. During normal operation, the monitoring loop executed consistently within the 2-second cycle, all three user roles authenticated correctly with appropriate permission enforcement, and the ML anomaly detector completed its warm-up phase and began generating scores after approximately two minutes of operation.

The dashboard provided real-time updates without lag, and all modules functioned accurately, ensuring smooth user interaction and system stability.

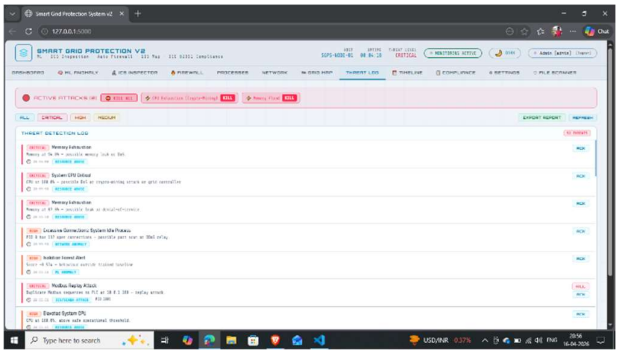

Figure 7.2.Threat Detection Alerts Tab

Figure 7.2 Threat Detection Alerts Tab shows a Under simulated attack conditions, the CPU Exhaustion and Memory Flood attacks were detected within 2–4 seconds of launch, generating CRITICAL severity alerts that appeared on the dashboard in real time. The firewall block action successfully added iptables DROP rules for identified threat source IPs within the same monitoring cycle as initial detection. The alert notification system promptly triggered email and SMS alerts to the configured recipients. System logs accurately recorded all attack events and response actions for further analysis. Overall, the system demonstrated fast detection and effective mitigation during attack scenarios.

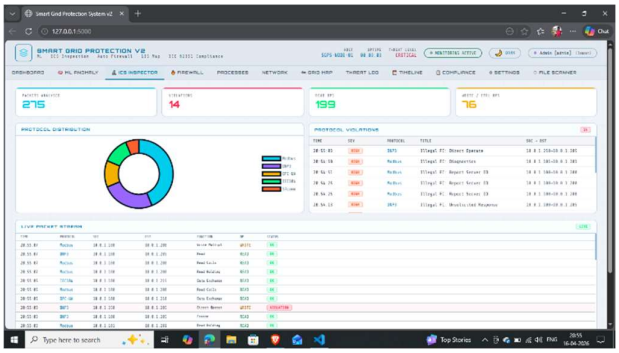

Figure 7.3 ICS Inspector Tab

Figure 7.3 – ICS Inspector Tab shows displayed live simulated protocol traffic across all six industrial protocols. During the Modbus DoS and DNP3 Replay attacks, the inspection engine correctly identified illegal function codes, unsolicited DNP3 responses, and unauthorised master IP addresses, generating

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 4 | April-2026
ISSN (O) 3107-6696

protocol violation records that appeared in real time on the dashboard. The system effectively differentiated between normal and malicious protocol behavior. All detected violations were logged with detailed packet-level information for analysis. Real-time inspection ensured immediate visibility of protocol anomalies. Overall, the module strengthened protocol-level security monitoring within the system.

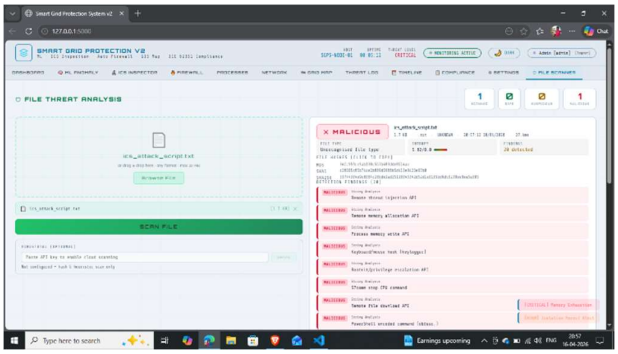

Figure 7.4. File Scanner Tab

Figure 7.4. File Scanner Tab shows successfully classified all eight test case files with the expected results. The EICAR test file was detected by hash match in Stage 1. A high-entropy random binary file was flagged as MALICIOUS in Stage 3. A PE executable with a .txt extension was detected as MALICIOUS due to the extension/magic mismatch check in Stage 2. All six scanner stages completed in under 50 milliseconds for files up to 5 MB. The scanning process demonstrated high accuracy across different types of threats and file conditions. Each stage of analysis contributed effectively to the final classification result. The system maintained fast processing speed without impacting overall performance. Detailed scan reports provided clear insights for further investigation and validation.

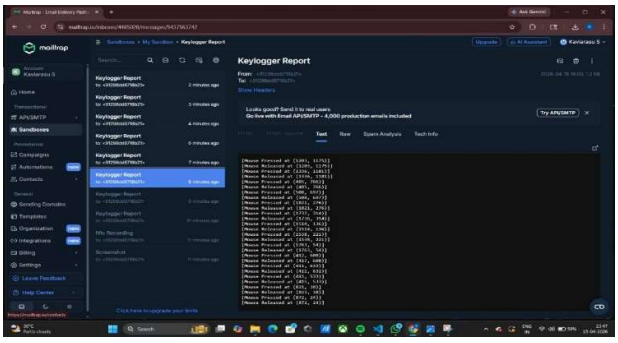

Figure 7.5. Keylogger Email Report Output

Figure 7.5. Keylogger Email Report Output shows the keylogger report received through the Mailtrap email service. It contains detailed logs of user activities such as mouse clicks and movements. The continuous logging of user interactions proves that the spyware is actively capturing input data and sending it externally, simulating a real-world data exfiltration attack.

The timely delivery of the report confirms successful alert and notification functionality. The captured data demonstrates the potential risk to user privacy and system security. This scenario validates the system's capability to detect and report spyware-related activities effectively.

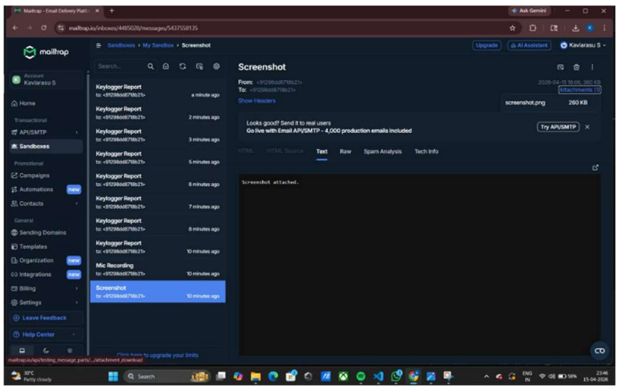

Figure 7.6 Screenshot Data Transmission

Figure 7.6 Screenshot Data Transmission shows This screenshot shows an email containing images captured by the spyware. The message confirms that screen data has been successfully captured and transmitted. This highlights the spyware's ability to take screenshots of user activity and send sensitive visual information externally.

This behavior demonstrates the risk of exposing confidential on-screen data. The system's detection mechanisms can help identify such unauthorized screen capture activities. It emphasizes the importance of protecting user sessions and sensitive information from spyware threats.

VII CONCLUSION AND FUTURE ENHANCEMENT

The Smart Grid Protection System (SGPS v3) successfully provides real-time cybersecurity monitoring and automated response for smart grid and ICS environments. It integrates multiple detection techniques to ensure accurate and efficient threat identification. The system demonstrated strong performance during both simulated and real-world attack scenarios.

Future enhancements can focus on improving scalability, integrating advanced machine learning models, and enabling real-time deployment in actual grid environments. Additional features such as secure communication, enhanced compliance mapping, and database integration can further strengthen the system. Overall, SGPS v3 serves as a solid foundation for developing advanced and practical cybersecurity solutions.

The Smart Grid Protection System v3 provides a comprehensive, real-time cybersecurity monitoring and automated response platform for smart grid and ICS environments. It integrates multiple detection layers, including rule-based detection, machine learning–based anomaly detection, ICS packet inspection, and file threat analysis, into a unified web-based dashboard. All simulated attacks were successfully detected within the monitoring cycle, and automated responses such as firewall blocking and alert generation operated effectively. The Isolation Forest model efficiently identified abnormal system behavior without relying on heavy external libraries, ensuring lightweight and scalable implementation. The compliance module mapped detected threats to IEC 62351 and NERC CIP standards, enabling structured evaluation of the system's security posture. This demonstrates that a software-based solution can provide effective cybersecurity protection without the need for expensive hardware infrastructure.

Furthermore, the system was validated using a real spyware scenario, where SGPS successfully detected suspicious activities such as continuous logging, unauthorized access, and data transmission. The system responded in real time with alerts and mitigation actions, proving its capability to handle real-world cyber threats. In conclusion, SGPS v3 is a reliable, efficient, and scalable cybersecurity solution that bridges the gap between academic research and practical implementation, providing a strong foundation for advanced smart grid protection and automated threat response.

Future enhancement involves enabling secure communication through HTTPS using TLS 1.3, along with automatic certificate renewal mechanisms to protect sensitive data in transit. The system can also be improved by introducing persistent storage for the Isolation Forest model, allowing it to retain learned behavior and eliminate the need for repeated warm-up phases after each restart. Furthermore, integrating a relational database such as PostgreSQL or SQLite would replace the current in-memory and JSON-based storage, enabling efficient data management, long-term storage, and advanced reporting capabilities. The compliance module can be made more dynamic by evaluating IEC 62351 controls based on real-time system configurations and observed security conditions rather than static assessments. In addition, incorporating MITRE ATT&CK for ICS framework mapping will allow better classification and correlation of detected threats, providing deeper insights for security analysis. Overall, these enhancements will make SGPS more secure, scalable, and suitable for deployment in real-world industrial cybersecurity environments.

REFERENCES

1. K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security," NIST Special Publication 800-82 Revision 2, National Institute of Standards and Technology, 2015.

2. G. Liang, J. Zhao, F. Luo, S. Weller, and Z. Dong, "A Review of False Data Injection Attacks Against Modern Power Systems," IEEE Transactions on Smart Grid, vol. 8, no. 4, pp. 1630–1638, 2017.

3. S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber-Physical System Security for the Electric Power Grid," Proceedings of the IEEE, vol. 100, no. 1, pp. 210–224, Jan. 2012.

4. Y. Liu, M. Reiter, and P. Ning, "False Data Injection Attacks Against State Estimation in Electric Power Grids," ACM CCS 2009, pp. 21–32, 2009.

5. F. Tao Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), pp. 413–422, 2008.

6. C. Ten, C. Liu, and G. Manimaran, "Vulnerability Assessment of Cybersecurity for SCADA Systems," IEEE Transactions on Power Systems, vol. 23, no. 4, pp.

1836–1846, 2008.

7. I. Fovino, M. Masera, and A. De Cian, "Integrating Cyber Attacks Within Fault Trees," Reliability Engineering and System Safety, vol. 94, no. 9, pp. 1394–1402, 2009.

8. R. Langner, "Stuxnet: Dissecting a Cyberweapon," IEEE Security and Privacy, vol. 9, no. 3, pp. 49–51, 2011.

9. A. Cherepanov, "WIN32/INDUSTROYER: A New Threat for Industrial Control Systems," ESET Research White Paper, 2017.

10. Dragos Inc., "TRISIS Malware: Analysis of Safety System Targeted Attack," Dragos Research, 2017.

11. H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, "Cybersecurity of Industrial Cyber-Physical Systems: A Review," ACM Computing Surveys, vol. 54, no. 11s, pp. 1–35, 2022.

12. I. A. Tøndel, M. G. Jaatun, and P. H. Meland, "Security Requirements for the Rest of Us: A Survey," IEEE Software, vol. 25, no. 1, pp. 20–27, 2008.

13. NERC, "Critical Infrastructure Protection (CIP) Standards," North American Electric Reliability Corporation, Version 6, 2016.

14. IEC 62351, "Power Systems Management and Associated Information Exchange — Data and Communications Security," Parts 1–12, International Electrotechnical Commission, 2007–2021.