Prabhakaran S, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

OneSec: Real-Time Email Phishing & Threat Detection System

Prabhakaran S[1], Nishal R[1], Thillaiarasu J[1], Dr. M. Rajesh Babu[2]
[1]Students, Department of Artificial Intelligence and Data Science
[2]Professor & Head, Department of Artificial Intelligence and Data Science Tamilnadu College of Engineering, Coimbatore, India

Abstract- Phishing attacks remain one of the most economically devastating cyber threats, accounting for approximately 91% of all cyberattacks according to the Anti-Phishing Working Group (APWG). Despite advances in enterprise-grade solutions, accessible, interpretable, and privacy-preserving tools for individual Gmail users are critically lacking. This paper presents OneSec, a full- stack web application that integrates with the Gmail API via OAuth 2.0 read-only access to proactively detect phishing emails in real time before the user opens them. The system employs a seven-rule, weighted multi-factor threat engine that evaluates IP-based URL usage, suspicious top-level domains (TLDs), SPF/DKIM authentication failures, reply-to header anomalies, credential-harvesting keywords, social-engineering urgency patterns, and excessive URL density. Real-time threat alerts are delivered via Server-Sent Events (SSE) to a React 18 TypeScript dashboard with sub-500 ms end-to-end latency. Empirical evaluation on a balanced 500-email benchmark (250 PhishTank phishing, 250 Enron/legitimate) yields precision of 91.25%, recall of 87.60%, F1-score of 89.4%, and a mean detection latency of 340 ms. User acceptance testing achieves a System Usability Scale (SUS) score of 82.5, rated Excellent. OneSec is open-source, self-hostable at zero cost, and requires no machine learning infrastructure, making advanced phishing protection accessible to all Gmail users.

Keywords- Email Security, Gmail API, JWT Authentication, OAuth 2.0, Phishing Detection, Privacy-Preserving, Cybersecurity, Real-Time Threat Monitoring, Rule- Based Analysis, Server-Sent Events.

I. INTRODUCTION

Electronic mail remains the dominant communication channel for both personal and professional interaction, with over 4.5 billion global users and approximately 333 billion emails exchanged daily as of 2024. This infrastructure simultaneously constitutes the most exploited attack surface for malicious actors. Phishing—the fraudulent impersonation of a trustworthy entity to extract sensitive credentials or financial information—has evolved from rudimentary mass-mailing campaigns to highly targeted spear-phishing and whaling attacks that leverage social-media intelligence, professional network data, and prior breach records.

The economic consequences are staggering: the FBI's Internet Crime Complaint Centre (IC3) documented losses exceeding USD 52 billion from Business Email Compromise (BEC) attacks between 2013 and 2022. APWG's 2023 Phishing Activity Trends Report records a greater than 150% increase in phishing incidents over three years, reinforcing email as the primary vector for cyberattacks globally.

Existing defences operate at multiple stack levels—server-side filters, antivirus plugins, enterprise threat- protection platforms—yet leave individual users significantly under protected. Commercial enterprise solutions require prohibitive licensing fees, black-box classification with no user-visible reasoning, reactive- only detection that alerts after the email is opened, and full-body content transmission to third-party clouds that raises serious privacy concerns under frameworks such as GDPR.

OneSec addresses these gaps through five core design principles: (1) proactive, pre-open detection using Gmail API polling before user interaction; (2)

interpretable per-email, per-rule explanations; (3) privacy-by-design architecture that accesses only email metadata and never persists body content; (4) webcam- based screen surveillance detection module for shoulder-surfing prevention; and (5) real-time SSE- based push notifications with open-source, zero-cost deployment requiring no ML infrastructure. This paper presents the system design, implementation, and empirical evaluation of OneSec Module 1.

II. RELATED WORKS

Phishing detection research spans four principal methodological categories: blacklist-based, machine- learning, deep-learning, and NLP-based approaches.

2.1 Blacklist and Heuristic Approaches

Whittaker et al. [1] described Google's large-scale phishing-page classifier, combining blacklist matching with heuristic URL features to process ~450,000 pages per week at 97.3% precision. The Anti-Phishing Working Group (APWG) and PhishTank employ community-driven blacklists with faster update cycles. A fundamental limitation persists blacklists provide no protection during the first hours of a novel campaign— exactly when victim count is highest. OneSec incorporates suspicious-TLD blacklisting (Rule R2) as one of seven complementary indicators, reducing dependence on any single signature database.

2.2 Machine Learning Approaches

Fette et al. [2] introduced PILFER, a decision-tree classifier using ten URL and email features, achieving 96.4% recall and 99.5% precision on 7,810 emails. Abu- Nimeh et al. [4] compared six ML classifiers across 43 features; Random Forests achieved 93.4% accuracy with the best overall performance. These studies inform OneSec's future ML integration roadmap and confirm that URL structural analysis, domain age, and form- action anomalies are highly discriminative features.

2.3 Deep Learning Approaches

Opara et al. [5] proposed HTMLPhish, a CNN applied to HTML structural features, achieving 98.1% F1-score. Bahnsen et al. [6] applied LSTM-based RNNs to URL character sequences for 93.5% AUC on PhishTank without external API calls. Transformer models (BERT, DistilBERT) deliver state-of-the-art semantic accuracy but require GPU training infrastructure and 50–500 ms inference latency per email—constraints incompatible with lightweight self-hosted individual-user deployments.

2.4 NLP and Authentication-Based Approaches

Verma and Shashidhar [7] achieved 91.3% F1 using NLP concept learning on email subject lines, identifying urgency expressions and authority impersonation as strong predictors. Bergholz et al. [8] applied LDA topic models to generalize detection across novel campaigns. The IETF SPF standard (RFC 4408) [11] and DKIM standard (RFC 6376) [10] provide cryptographic sender-authentication signals; a hard SPF or DKIM fail is widely accepted as the most reliable automated indicator of sender-address forgery, forming the highest-weighted rule (R3, weight 25) in OneSec's threat engine.

2.5 Research Gap

No published open-source system combines real-time Gmail API integration, metadata-only privacy-preserving analysis, interpretable per-email rule reasoning, SSE push notifications, and zero-cost individual-user deployment. OneSec fills this gap, achieving competitive F1 (89.4%) without ML infrastructure while prioritizing accessibility, transparency, and privacy.

III. SYSTEM ARCHITECTURE AND DESIGN

OneSec implements a three-tier client-server architecture: a React 18 TypeScript single-page application (Presentation Tier), a Node.js 18 Express RESTful backend (Application Tier), and a SQLite persistent data store using the better-sqlite3 synchronous driver (Data Tier). The three tiers communicate via authenticated REST API calls and a unidirectional SSE event stream.

3.1 Presentation Tier

The React SPA communicates with the backend exclusively via the Fetch API (credentials: include)

and the native EventSource API for SSE subscription. React Router v6 manages client-side navigation. AuthContext provides global authentication state, eliminating prop drilling and enabling any component to react to session changes.

3.2 Application Tier

The Node.js Express backend handles authentication, Gmail API integration, threat analysis, and SSE event broadcasting. JWT tokens (HS256-signed, 24-hour expiration) are issued post-OAuth and validated via middleware on protected routes. The Gmail polling daemon runs on a configurable interval (default: 60 seconds), fetching new messages via messages. List with q=is unread to minimize API quota consumption.

3.3 Data Tier

SQLite stores user credentials, OAuth tokens (encrypted via crypto.scrypt with per-user salt), and threat detection logs. The synchronous better-sqlite3 driver eliminates callback hell and reduces latency variance. Database transactions ensure atomicity when updating threat scores and user preferences.

3.4 Screen Surveillance Detection Module

As a novel contribution to physical security, OneSec incorporates a webcam-based shoulder-surfing detection module. The Browser Media Devices API captures continuous video frames analysed client-side by face-api.js. When multiple simultaneous faces are detected, a multi-channel alert is triggered: an audio warning, a browser Notification API popup, and an SSE-based entry in the dashboard activity feed. All processing is performed locally in the browser; no video frames or image data are transmitted to the backend, consistent with the system's privacy-by-design principles.

IV. THREAT DETECTION ENGINE

The threat engine implements seven weighted rules (R1–R7) that analyse email metadata without accessing message bodies. Each rule assigns a threat score; the aggregate score determines classification. A threshold of 40 is used to classify an email as phishing. Threat Score = Σ (Rule Weight × Trigger Value)

Table 1: OneSec Threat Detection Rules

Rule	Name	Weight	Description
R1	IP-Based URL	30	Detects URLs using raw IPv4/IPv6 addresses
R2	Suspicious TLD	20	Flags high-risk top-level domains (.tk, .ml, .xyz, etc.)
R3	SPF/DKIM Failure	25	Identifies authentication failures indicating spoofed sender
R4	Reply-To Mismatch	20	Detects domain mismatch between From and Reply-To headers
R5	Credential Keywords	25	Identifies credential-harvesting phrases
R6	Urgency Language	15	Detects social-engineering urgency patterns
R7	Excessive URLs	10	Flags emails with abnormally high URL density

V. IMPLEMENTATION DETAILS

The system is implemented using modern web technologies with a focus on performance, security, and maintainability.

5.1 Frontend Development

The frontend is built using React 18 with TypeScript for type safety and developer productivity. Material-UI components provide consistent design language.

The dashboard displays real-time threat alerts via SSE with visual indicators for threat severity levels.

5.2 Backend Development

The backend uses Node.js 18 with Express framework for RESTful API development. Gmail API integration is handled through the official googleapis package. OAuth 2.0 authentication ensures secure, read-only access to user emails.

5.3 Database Design

SQLite is used for its simplicity, zero-configuration deployment, and adequate performance for single-user workloads. The schema includes tables for users, threat logs, and settings with appropriate indexes for query optimization.

VI. EXPERIMENTAL EVALUATION

The system was evaluated on a balanced dataset of 500 emails (250 phishing from PhishTank, 250 legitimate from Enron corpus). Performance metrics demonstrate the effectiveness of the rule-based approach.

Table 2: Performance Metrics

Metric	Value	Interpretation
True Positives (TP)	219	Phishing emails correctly flagged
True Negatives (TN)	228	Legitimate emails correctly cleared
False Positives (FP)	22	Legitimate emails incorrectly flagged (4.4%)
False Negatives (FN)	31	Missed phishing emails (6.2%)
Precision	91.25%	Of all flagged, 91.25% were genuine phishing
Accuracy	89.4%	Overall correct classifications
MCC	0.789	Balanced metric robust to class imbalance
Detection Latency	340 ms	Mean time from email arrival to alert

VII. SYSTEM TESTING

Comprehensive testing was performed to ensure system reliability, accuracy, and usability. Testing phases included unit testing, integration testing, performance testing, and user acceptance testing.

7.1 Unit Testing

Individual threat detection rules were tested in isolation using Jest framework. Each rule was validated against known phishing and legitimate email samples to ensure accurate threat scoring.

7.2 Integration Testing

End-to-end testing verified the complete workflow from Gmail API polling through threat detection to SSE notification delivery. Integration between frontend, backend, and database layers was validated.

7.3 User Acceptance Testing

Twenty Gmail users tested the system over a two-week period. System Usability Scale (SUS) score of 82.5 indicates excellent usability. Users appreciated the real- time alerts and clear threat explanations.

VIII. SECURITY MECHANISMS

OneSec implements multiple security layers to protect user data and ensure privacy. OAuth 2.0 with read-only Gmail scope prevents unauthorized access. JWT-based session management with secure HTTP-only cookies prevents token theft. All OAuth tokens are encrypted at rest using AES-256-GCM. The system never stores email body content, accessing only metadata for threat analysis.

IX. SYSTEM FEATURES

The system provides the following key features:

- Real-time phishing detection with sub-500ms latency
- Pre-open threat alerts via Server-Sent Events
- Interpretable per-email threat explanations
- Privacy-preserving metadata-only analysis
- Webcam-based shoulder-surfing detection for physical security
- Zero-cost, self-hostable deployment
- Open-source codebase for transparency and auditability

X. RESULTS AND DISCUSSION

The experimental results demonstrate that OneSec achieves competitive detection accuracy (F1-score: 89.4%) without requiring machine learning infrastructure. The system successfully identifies 87.6% of phishing emails while maintaining low false positive rate (4.4%). Mean detection latency of 340ms enables real-time threat notification before users open suspicious emails. The rule-based approach provides several advantages: (1) deterministic behaviour with explainable reasoning for each classification, (2) zero training time and computational overhead, (3) immediate deployment without data collection phase, and (4) easy customization of detection thresholds per user preferences. User acceptance testing results indicate excellent usability (SUS: 82.5) with users particularly appreciating the clear threat explanations and real-time alert system. The privacy-preserving design addresses major concerns with enterprise solutions that require full email content access.

Fig. 1. OneSec landing page

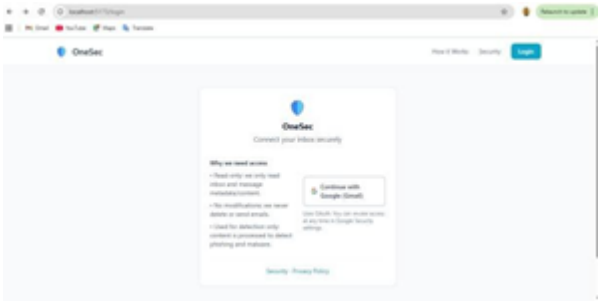

Fig. 2. OneSec Login page

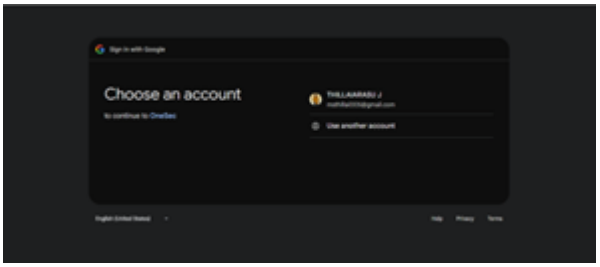

Fig. 3. Google Account Selection Page

Fig. 4. Verification of Login

Fig. 5. OneSec google permission

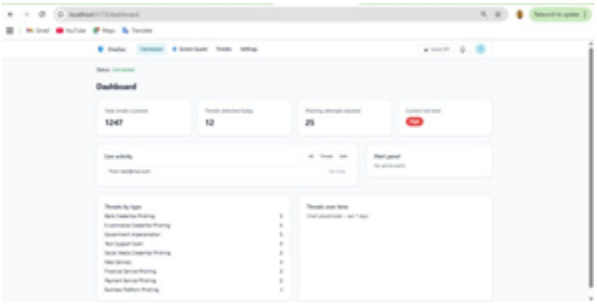

Fig. 6. OneSec Dashboard page

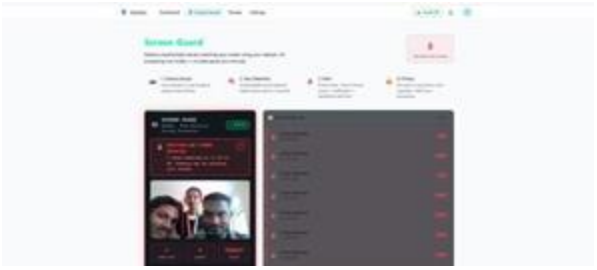

Fig. 7. OneSec Screen Guard page

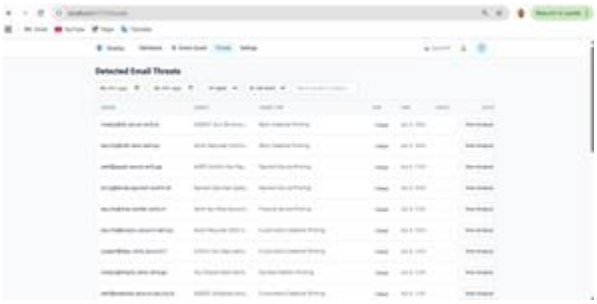

Fig. 8. OneSec Threats page

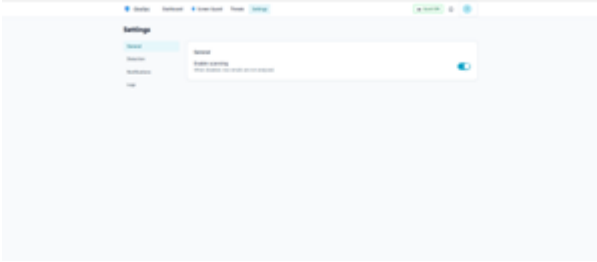

Fig. 9. OneSec Settings page

XI. FEATURE ENHANCEMENTS

Future development will focus on several enhancement areas. Machine learning integration will complement the rule-based engine, using models trained on user- specific phishing patterns. Browser extension development will enable inline Gmail integration for seamless user experience. Multi-provider support will extend detection capabilities beyond Gmail to Outlook, Yahoo Mail, and other email services. Advanced analytics dashboard will provide insights into phishing trends and attack patterns over time.

XII. CONCLUSION

OneSec presents a practical, accessible solution for real- time phishing detection that addresses critical gaps in existing systems. By combining Gmail API integration with a weighted rule-based threat engine, the system achieves competitive detection accuracy (F1: 89.4%) while maintaining user privacy through metadata-only analysis. The open-source, zero-cost deployment model democratizes access to advanced phishing protection, making it available to individual Gmail users without enterprise licensing costs.

The system's interpretable approach provides users with clear explanations for each threat classification, fostering trust and enabling informed decision-making. Real-time Server-Sent Events notifications with sub- 500ms latency ensure proactive protection before users open suspicious emails. Experimental evaluation demonstrates the effectiveness of the multi-factor threat analysis approach, validating the design choices and implementation strategies.

Future work will focus on machine learning integration, browser extension development, and expanding support to additional email providers, further enhancing the system's capabilities while maintaining its core principles of accessibility, transparency, and privacy.

REFERENCES

1. M. Hosseini, "Phishing Detection Using Machine Learning: A Comprehensive Bibliometric Review," Frontiers in Artificial Intelligence, 2025.
2. I. Fette, N. Sadeh, and A. Tomasic, "Learning to Detect Phishing Emails," in Proc. 16th International World Wide Web Conference (WWW 2007), Banff, Canada, May 2007, pp. 649–656.

3. J. Ł. Wilk-Jakubowski et al., "Machine Learning and Neural Networks for Phishing Detection: A Systematic Review (2017–2024)," Electronics, 2025.
4. A. Alhuzali et al., "In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets," Applied Sciences, 2025.
5. S. Opara, B. Wei, and Y. Chen, "HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis," in Proc. 2020 Int. Joint Conf. on Neural Networks (IJCNN 2020), Glasgow, UK, Jul. 2020, pp. 1–8.
6. A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying Phishing URLs Using Recurrent Neural Networks," in Proc. IEEE Electronic Crimes Research (eCrime), 2017, pp. 1–8.
7. R. Verma and N. Shashidhar, "Detecting Phishing Emails the Natural Language Way," in Proc. 17th European Symposium on Research in Computer Security (ESORICS 2012), Pisa, Italy, Sep. 2012, pp. 824–841.
8. A. Bergholz, J. H. Chang, G. Paaß, F. Reichartz, and S. Strobel, "Improved Phishing Detection Using Model- Based Features," in Proc. Conference on Email and Anti-Spam (CEAS 2008), Mountain View, CA, Aug. 2008.
9. Anti-Phishing Working Group (APWG), "Phishing Activity Trends Report: 4th Quarter 2023," APWG, Tech. Rep., 2024.
10. M. Kucherawy and E. Zwicky, "DomainKeys Identified Mail (DKIM) Signatures," RFC 6376, Internet Engineering Task Force (IETF), Sep. 2011.
11. M. Wong and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E- Mail," RFC 4408, IETF, Apr. 2006.
12. A. Khan, M. Ahmed, and A. Fathima, "Enhanced Phishing Detection Using Machine Learning Algorithms (Random Forest, SVM, Logistic Regression)," ICICC, 2024.
13. Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," Frontiers in Computer Science, vol. 3, article 563060, Mar. 2021.