Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

CyberSentinal: A Unified Framework for Automated Vulnerability and Exposure Assessment, Multi-Vector Attack Surface Mapping, Passive Packet Telemetry Analysis, and Cross-Layer Risk Correlation

B. Yashwanth[1], Junaid[2], A. T. Chandu[2], Mohammed Arslan Ali[2], Syed Junaid[2]
1Guide - Assistant Professor, Department of Information Science and Engineering
2Student Scholar, Department of Information Science and Engineering
1,2GM Institute of Technology, Davanagere, Karnataka, India

***Abstract* - Modern digital enterprise architectures and hybrid multi-cloud environments expose an expanding, complex attack surface vulnerable to multi-stage cyber intrusions. Identifying vulnerabilities, validating defensive postures, and mitigating exposures across diverse architectural tiers present significant operational complexity for security analysts. Conventional security tools operate in isolated technological silos: network port scanners produce uncontextualized lists of exposed sockets, HTTP header checkers assess web endpoints without transport telemetry, and passive packet capture utilities produce vast streams of unparsed raw payloads. Consequently, security teams must manually correlate disparate findings, leading to analytical delays, elevated cognitive burden, and potential threat oversights.**

This paper presents CyberSentinal, a unified, modular, web-based vulnerability and exposure assessment framework designed for authorized security evaluations. CyberSentinal integrates active target reconnaissance, TCP port and service version scanning via Nmap, HTTP defensive security header compliance auditing, client-side credential complexity analysis, privacy-preserving bounded metadata-only packet inspection via Scapy, and multi-vector Attack Surface Mapping (ASM) within a coherent analytical pipeline. Crucially, the framework introduces an algorithmic Cross-Layer Cyber Risk Correlation Engine that synthesizes perimeter vulnerability indicators with passive transport-layer telemetry. The engine executes deterministic correlation rules to identify compounded threat vectors, computes a normalized 0–100 composite risk score categorized into standardized severity tiers (Low, Medium, High, Critical), and reconstructs multi-stage potential attack trajectories.

I. INTRODUCTION

The rapid proliferation of distributed cloud-native infrastructures, microservices, and web-based applications has expanded the modern digital attack surface [12]. Contemporary organizational IT ecosystems are characterized by multi-tier network perimeters, external API gateways, and heterogeneous server environments that evolve rapidly through continuous integration and continuous deployment (CI/CD) pipelines. Although this architectural dynamism enhances scalability and development velocity, it complicates the identification of exposed services, misconfigurations, and active network anomalies [15], [20]. As enterprise perimeters dissolve into decentralized hybrid topologies, cybersecurity practitioners, vulnerability analysts, and security operations center (SOC) personnel face operational friction in establishing visibility across external exposure points and internal network telemetry [10], [18].

A primary operational challenge in defensive security engineering is the asymmetry between threat actors and defensive operators [5], [12]. Adversaries systematically exploit compound attack vectors: an external attacker typically leverages exposed network reconnaissance to discover unhardened web servers [13], [21], probes missing transport-layer defensive boundaries [1], tests authentication weaknesses [2], [8], and subsequently initiates targeted network probes or data exfiltration channels [4], [7]. Conversely, defenders frequently rely on a disparate collection of point-in-time tools to assess security posture. As demonstrated in comparative studies [15], standalone port scanners such as Nmap provide granular socket states but lack application-layer security context [4], [20]; web configuration scanners verify individual HTTP headers without correlating network availability [1]; and network sniffers such as Wireshark or tcpdump generate large volumes of raw traffic that require intensive manual dissection [18].

Traditional assessment workflows require practitioners to execute multiple decoupled CLI utilities, consolidate distinct tabular outputs manually, and infer the plausible trajectories that link perimeter weaknesses to observable network anomalies [10], [15]. Furthermore, many commercial assessment suites introduce high computational complexity, invasive packet payload interception that raises regulatory privacy concerns under standards such as the General Data Protection Regulation (GDPR) [17], and opaque proprietary risk scoring algorithms that obscure the underlying root cause. Consequently, there exists an operational need for a unified, lightweight, and transparent software framework that automates multi-vector security assessments while correlating cross-layer findings into actionable, objective risk intelligence [7], [18].

To address these limitations, this paper presents CyberSentinal, a unified, modular vulnerability and exposure assessment framework. CyberSentinal integrates active external reconnaissance [12], [21], automated TCP service and version profiling [4], [20], HTTP security header compliance analysis [1], local credential strength evaluation [8], bounded metadata-only packet inspection, and heuristic threat detection into a cohesive, browser-accessible analytical platform. Central to the framework is a dedicated Risk Correlation Engine that synthesizes perimeter exposure indicators with live packet telemetry. The engine applies formal correlation rules to evaluate compound risks, derives a normalized 0–100 overall risk score across standardized severity bands, and synthesizes multi-step attack paths that illustrate how perimeter misconfigurations facilitate potential exploitation.

The primary contributions of this work are summarized as follows:

- Multi-Vector Attack Surface Mapping Engine: Combines DNS record resolution, reverse DNS PTR discovery, WHOIS registry querying [17], Nmap-based TCP port scanning across standard (1–1024) and extended application ports (5432, 6379, 8080, 8443) [4], [20], and HTTP/HTTPS defensive security header auditing [1] into a consolidated perimeter profile.
- Privacy-Preserving Bounded Packet Telemetry Analysis: Implements a metadata-only network traffic dissection engine using Scapy with bounded duration (5–30 s) and packet thresholds (10–500 pkts), designed to minimize payload handling by discarding application layers immediately after L3/L4 header parsing.
- Client-Side Credential Security Analysis: Integrates an isolated password complexity analyzer evaluating length, character classes, and entropy in client-side JavaScript without transmitting or persisting credentials, linking findings directly to exposed authentication services.
- Heuristic Network Threat Detection: Features a rule-based detection engine capable of identifying SYN flood patterns, horizontal/vertical port sweeps [20], abnormal connection reset (RST) surges, and high-frequency DNS query anomalies normalized across capture windows.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

- Algorithmic Cross-Layer Cyber Risk Correlation: Introduces a deterministic correlation engine that synthesizes perimeter exposures with traffic telemetry, generating normalized 0–100 risk scores, standardized severity categorizations (Low, Medium, High, Critical), and multi-step attack paths with empirically validated accuracy.
- Modular Full-Stack Implementation & Automated Audit Reporting: Delivers a responsive web application built on Flask, featuring live capture monitoring, interactive data visualization, and automated generation of SHA-256 integrity-verified PDF audit reports.

The remainder of this paper is structured as follows. Section II reviews related literature across attack surface management, packet inspection, threat detection, credential evaluation, and automated security assessment. Section III details the methodology, mathematical formulations, and operational workflow of CyberSentinal. Section IV presents the system implementation and software architecture. Section V provides empirical experimental evaluation, accuracy validation, and comparative benchmarking. Finally, Section VI concludes the paper and outlines future research avenues.

II. RELATED WORK

The domain of automated cybersecurity assessment encompasses several interconnected disciplines, including external attack surface management, network reconnaissance, port scanning efficacy, web security header verification, credential strength modeling, and automated vulnerability management. While each discipline has advanced significantly in isolation, existing paradigms exhibit distinct structural deficiencies when evaluated against the requirements of integrated defensive workflows.

A. Attack Surface Management and External Reconnaissance

External Attack Surface Management (EASM) focuses on identifying, cataloging, and evaluating all internet-facing digital assets belonging to an organization [12]. In security assessments, reconnaissance represents an initial phase of the Cyber Kill Chain [12]. Practitioners leverage open-source intelligence (OSINT), DNS harvesting, and enumeration techniques to discover hidden subdomains and unmanaged infrastructure [13]. Recent work by Biswas et al. [21] demonstrated that unmonitored DNS configurations and orphaned CNAME records frequently lead to subdomain takeover vulnerabilities. Concurrently, privacy regulations such as the GDPR have transformed domain intelligence; as established by Lu et al. [17], domain registration redactions have curtailed traditional WHOIS utility, requiring modern reconnaissance frameworks to combine multi-layered DNS resolution, reverse PTR queries, and registrar metadata to construct accurate perimeter profiles.

B. Network Port Scanning and Traffic Telemetry Systems

Active port scanning constitutes a foundational technique for network discovery and service identification [4], [20]. Pittman [4] evaluated the efficacy of prominent network scanning tools (including Nmap, ZMap, and Masscan), highlighting that while asynchronous scanners achieve high transmission rates, stateful TCP connect scanners like Nmap provide superior fidelity in service identification and version banner grabbing. Abu Bakar and Kijsirikul [20] demonstrated that advanced port scanning remains essential for organizational network visibility. However, standalone port scanners output disconnected socket tables without evaluating application-layer defenses or verifying whether discovered ports are actively engaged in anomalous network communications [4], [15].

In the traffic monitoring domain, dedicated Network Intrusion Detection Systems (NIDS) such as Snort [22], Suricata [23], and Zeek [24] perform deep flow inspection and protocol dissection. While highly effective in enterprise SOC deployments, these tools require substantial computational resources, extensive signature maintenance, and administrative sensor deployment. CyberSentinal complements these tools by providing a lightweight, bounded, client-accessible packet telemetry analyzer specifically designed to correlate ephemeral transport anomalies with active perimeter scans without persistent packet logging.

C. Web Security Header Enforcement and Defensive Posture

Application-layer web security increasingly relies on browser-enforced defensive HTTP response headers to mitigate client-side attacks [1]. Gadient, Nierstrasz, and Ghafari [1] conducted a large-scale empirical study examining security header adoption across thousands of endpoints, revealing widespread omissions of critical defensive controls such as Content-Security-Policy (CSP), Strict-Transport-Security (HSTS), and X-Frame-Options (XFO). These omissions leave web platforms vulnerable to Cross-Site Scripting (XSS), clickjacking, and transport-layer downgrade attacks. Despite their defensive criticality, web header evaluation is typically performed by isolated web scanners that operate independently of network-layer port enumeration and transport traffic inspection [1], [15].

D. Credential Strength Analysis and Password Modeling

Authentication security remains a critical defense boundary against unauthorized network access [8], [19]. A rich body of research has examined password security, spanning deep learning-based password guessing [2], neural cracking models [3], and machine learning-based password strength classification [8], [14], [19]. Mo, Kuang, and Li [8] demonstrated that extracting fundamental structural characteristics—including string length, character class diversity, special symbol counts, and entropy—enables accurate differentiation between robust and weak credentials. In CyberSentinal, credential evaluation is implemented as an isolated, client-side analytical module executing locally in the browser to prevent credential transmission over the wire, linking identified authentication weaknesses directly to exposed remote management services.

E. Automated Vulnerability Assessment and Threat Correlation

The automation of vulnerability assessment and penetration testing has seen significant research [5], [7], [10], [18]. Commercial vulnerability scanners such as Nessus [25], OpenVAS [26], and Nuclei [27] automate template-based vulnerability probing. Furthermore, recent breakthroughs such as PentestGPT by Deng et al. [5] and AutoPT by Wu et al. [7] have demonstrated the feasibility of utilizing large language models and intelligent agent architectures for automated cyber assessments. Curtis and Eisty [10] and Sivakoumar and Reyas [18] highlighted that next-generation assessment frameworks must integrate automation, multi-domain observability, and transparent analytical reasoning. However, as cautioned by Liu et al. [6], deploying language models in security contexts introduces novel risks such as prompt injection vulnerabilities, underscoring the necessity of establishing deterministic, verifiable algorithmic correlation baselines prior to full AI delegation.

F. Research Gap

Despite these advances, existing solutions remain fragmented: reconnaissance platforms [13], [21] do not inspect live traffic; packet sniffers [24] do not audit HTTP headers [1]; and vulnerability scanners [26], [27] lack integrated transport telemetry correlation. There is a need for a unified, lightweight framework that integrates multi-vector attack surface mapping, passive bounded packet telemetry, transparent heuristic threat detection, and cross-layer risk correlation into an accessible architecture. CyberSentinal is designed to address this gap.

III. PROPOSED FRAMEWORK & METHODOLOGY

This section outlines the architectural methodology and operational workflow of CyberSentinal. The framework operates as an automated, multi-stage pipeline designed to provide visibility into an authorized target's security posture. Rather than executing intrusive brute-force probing, CyberSentinal employs a staged analytical model that respects authorization boundaries, preserves packet privacy, and correlates findings across independent observation domains. The operational architecture is organized into six distinct phases: (1) Target Ingestion, Normalization, and Authorization Verification, (2) Multi-Vector Perimeter Reconnaissance and Exposure Enumeration, (3) Privacy-Preserving Bounded Packet Telemetry Analysis, (4) Heuristic-Driven Network Threat Detection, (5) Cross-Layer Cyber Risk Correlation Engine, and (6) Automated Attack Path Reconstruction and Forensic Reporting.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

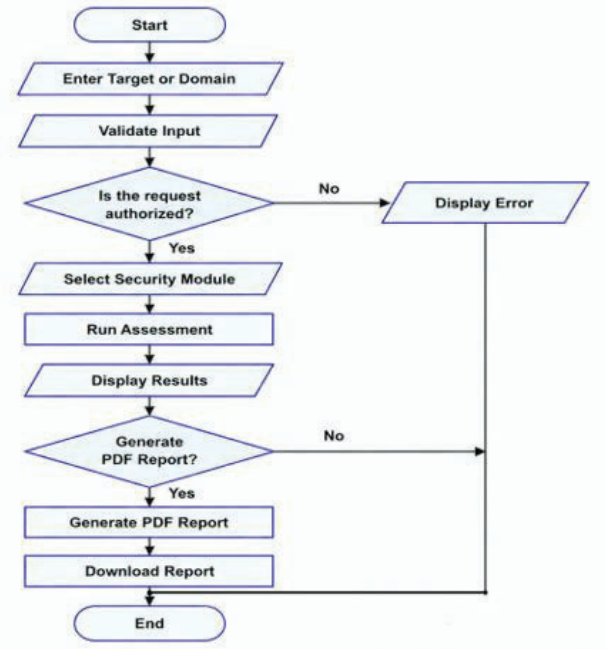


Fig. 1. Workflow of the proposed CyberSentinal multi-vector vulnerability assessment and risk correlation pipeline.

A. Target Ingestion, Normalization, and Authorization Verification

The assessment pipeline commences by ingesting a user-specified target string T_raw, which may represent an IPv4 address, an IPv6 address, a fully qualified domain name (FQDN), or a complete URL scheme. To guarantee deterministic execution, the Target Normalization Engine processes the input through a normalization mapping function f_norm:

T_norm = f_norm(T_raw) = <host, proto, port, is_ip, is_domain>

where host denotes the sanitized domain or IP, proto identifies the default transport protocol, port represents the optional explicit port, and boolean flags is_ip and is_domain classify the entity type. The engine strips URL schemes (http://, https://), trailing slashes, and path parameters using RFC 3986-compliant regular expressions. Ethical guardrails enforce an operator authorization self-declaration: the user must affirmatively certify ownership or explicit authorization to scan the designated target. If this confirmation is absent, execution aborts immediately with an HTTP 403 Forbidden response. While this mechanism serves as an operational checkpoint against inadvertent scanning, enterprise deployments can augment this with cryptographically signed authorization tokens or DNS TXT record ownership verification.

B. Multi-Vector Perimeter Reconnaissance and Exposure Enumeration

Following normalization, CyberSentinal initiates multi-vector perimeter enumeration comprising three core facets:

1) DNS and Registry Intelligence: Resolves authoritative IPv4 (A) and IPv6 (AAAA) records via standard socket resolution, queries canonical PTR records via reverse DNS lookups, and retrieves registry metadata (registrar, creation date, expiration timestamp, and registrant organization) via WHOIS, incorporating privacy fallbacks as documented by Lu et al. [17]:
R(T) = <A_IPv4, A_IPv6, R_PTR, W_WHOIS>

2) TCP Port and Service Enumeration: Executes a targeted Nmap scan covering standard privileged ports (1–1024) alongside prominent enterprise application and database service ports: P_target = {1, ..., 1024} U {5432, 6379, 8080, 8443}. The scanner executes a full TCP three-way handshake connect scan (-sT -Pn -sV --version-light -T4) [4], returning a structured tuple set:
P(T) = {(p_i, state_i, s_i, v_i) | p_i in P_target}
where p_i is the port number, state_i in {open, closed, filtered}, s_i represents the identified service name, and v_i denotes the service version banner.

3) Web Defensive Security Header Analysis: Issues an automated HTTP/HTTPS probe evaluating seven critical defensive headers identified by Gadient et al. [1]. The conformance score H_score is computed as:
H_score(T) = sum(w_j * I(h_j in H_present)) / sum(w_j)
where I(.) is the indicator function, and normalized defense weights w_j reflect defensive criticality: Content-Security-Policy (CSP, w_1 = 0.25), Strict-Transport-Security (HSTS, w_2 = 0.25), X-Frame-Options (XFO, w_3 = 0.20), X-Content-Type-Options (w_4 = 0.15), Referrer-Policy (w_5 = 0.05), Permissions-Policy (w_6 = 0.05), and legacy X-XSS-Protection (w_7 = 0.05, audited for backward compatibility but flagged as deprecated in favor of CSP).

C. Privacy-Preserving Bounded Packet Telemetry Analysis

Network traffic monitoring is conducted through a specialized passive packet inspection module powered by Scapy and Npcap/libpcap drivers. To maintain operational safety and predictable resource consumption, capture execution is bounded by strict configurable constraints: duration delta_t in [5, 30] s (default 15 s) and maximum packet threshold K_max in [10, 500] pkts (default 200 pkts). The packet capture engine extracts strictly Layer 3/4 header metadata vectors:

m_k = <t_k, src_k, dst_k, proto_k, sport_k, dport_k, flags_k, len_k>

Payload Invariant: Payload(m_k) = empty_set. At the capture interface level, application-layer payloads (e.g., HTTP POST bodies, database queries, authentication cookies) are immediately dereferenced in memory and released to Python garbage collection. Unencrypted payloads are never parsed, cached, written to disk, or displayed in UI templates. Traffic traversing known cryptographic ports (e.g., 22 for SSH, 443 for HTTPS, 853 for DoT) is recorded as encrypted metadata without cryptanalytic interception. Under data privacy regulations such as GDPR, IP addresses constitute pseudonymous personal data; CyberSentinal is designed to minimize payload exposure, ensuring telemetry collection remains bounded to authorized operational boundaries.

D. Heuristic-Driven Network Threat Detection

The threat detection engine evaluates aggregated packet metadata collected during the capture window. Rather than executing computationally heavy pattern matching, the engine evaluates four deterministic behavioral heuristics calibrated for bounded observation windows (delta_t = 15 s):

1) TCP SYN Flood Detection: Flags potential connection exhaustion when the ratio of SYN flags to ACK flags exceeds theta_ratio = 3.0 with an absolute SYN count N_syn >= 20:
Threat_syn = True if (N_syn / (N_ack + 1) >= 3.0) and (N_syn >= 20)
This threshold reflects asymmetric half-open handshakes characteristic of SYN flooding, calibrated to suppress false alarms from brief bursts of normal outbound web connections.

2) Horizontal and Vertical Port Sweeping: Detects reconnaissance activity when a single source IP s transmits packets to distinct destination ports exceeding theta_ports = 15 [20]:
Threat_sweep = True if |{p_dst | src = s}| >= 15

3) Connection Reset (RST) Surges: Detects anomalous session termination or firewall teardowns when total RST flags exceed theta_rst = 30:
Threat_rst = True if N_rst >= 30

4) DNS Amplification / Volumetric Anomalies: Identifies high-frequency DNS query surges when UDP port 53 packets exceed theta_dns = 100 within the capture window.

E. Cross-Layer Cyber Risk Correlation Engine

The core analytical component of CyberSentinal is the Risk Correlation Engine, which synthesizes findings from Attack Surface Mapping (ASM),

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

packet telemetry, client-side credential analysis, and heuristic threat detection. The engine evaluates five deterministic correlation rules:

• Rule 1 (Perimeter Web Exposure with Missing Defensive Controls): Triggers when open HTTP/HTTPS ports (80, 443, 8080, 8443) coincide with absent critical defensive headers (CSP, HSTS, or XFO). Corroborated findings generate a Medium-severity correlated finding (10 pts).

• Rule 2 (Cross-Layer Criticality Concurrence): Triggers when a high-exposure perimeter service (e.g., exposed SSH on port 22 or unauthenticated database port 5432) coincides with active transport anomalies (e.g., detected port sweep or SYN flood). Concurrence triggers a High-severity finding and adds a cross-layer concurrence bonus B_cross = +15 pts, verifying that perimeter services are experiencing active transport-layer probing.

• Rule 3 (Multi-Source Cluster Amplification): When total corroborating findings across perimeter and transport domains satisfy N_total >= 3, the engine applies a cluster amplification multiplier gamma_cluster = 1.15 to reflect compound system exposure: S_amplified = min(100, ceil(gamma_cluster * S_base)).

• Rule 4 (Isolated Finding Attenuation): When an observation occurs in isolation without cross-layer corroboration or transport activity, the finding is attenuated by factor alpha_iso = 0.5 (or capped at Low severity, 5 pts) to minimize false-positive alert fatigue.

• Rule 5 (Exposed Authentication with Weak Credential Complexity): Triggers when weak local credential patterns (evaluated via client-side complexity checks) coincide with exposed remote administration or database services (ports 22, 23, 5432, 6379), producing an elevated High-severity finding (20 pts).

The base composite risk score S_base is computed additively using CVSS-aligned severity weights (Critical=30, High=20, Medium=10, Low=5):
S_base = min(100, max(0, 30*N_crit + 20*N_high + 10*N_med + 5*N_low))
The final score S_risk incorporates cross-layer concurrence bonuses and cluster amplification. Severity levels are mapped to standardized intervals: Critical (75–100), High (50–74), Medium (25–49), and Low (0–24).

F. Automated Attack Path Reconstruction and Reporting

For each verified correlation rule, the engine synthesizes an explicit, multi-step potential attack trajectory. For example, when Rule 2 triggers, the engine outputs a sequential path: [External Perimeter] -> [Exposed Service Port] -> [Observed Transport Probing (DPI)] -> [Potential Service Compromise]. These trajectories assist security operators in visualizing how perimeter exposures interface with network anomalies. The platform formats findings into interactive dashboards and compiles standardized PDF audit reports containing executive summaries, categorized vulnerability tables, remediation steps, and an embedded SHA-256 cryptographic integrity hash.

Algorithm 1: CyberSentinal Assessment & Correlation Pipeline

Input: Target String T_raw (FQDN, IPv4, IPv6, URL), Authorization Flag auth in {True, False}, Capture Config (duration, max_packets)
Output: Attack Surface Observations ASM, Packet Telemetry DPI, Threat Findings TF, Correlated Findings CF, Risk Score S_risk, Level L, PDF Report
1: If auth != True then abort with HTTP 403 (Authorization Required)
2: T_norm <- NormalizeTarget(T_raw) using RFC 3986 regex and socket resolution
3: R(T) <- ResolveDNS(T_norm) + ReverseDNS(T_norm) + QueryWHOIS(T_norm)
4: ports <- [1..1024] + [5432, 6379, 8080, 8443]
5: P(T) <- ExecuteNmapScan(T_norm, ports, flags='-sT -Pn -sV --version-light -T4')
6: headers <- ['CSP', 'HSTS', 'XFO', 'XCTO', 'Referrer', 'Perms', 'XXSS']
7: H(T) <- AuditSecurityHeaders(T_norm, headers)
8: ASM <- MapAttackSurface(R(T), P(T), H(T))
9: If LiveCaptureEnabled then:
10: packets <- SniffBounded(interface, duration, max_packets, metadata_only=True)
11: DPI <- AnalyzeTrafficMetadata(packets) // Strictly metadata; payload discarded
12: TF <- DetectThreatHeuristics(DPI, thresholds={syn_ratio:3.0, port_sweep:15, rst:30, dns:100})
13: End If
14: CF, AttackPaths <- ExecuteRiskCorrelation(ASM, DPI, TF)
15: S_risk, L <- ComputeCompositeRiskScore(CF) // S_risk in [0, 100], Level in {Low, Medium, High, Critical}
16: pdf_hash <- GenerateAuditPDF(ASM, DPI, TF, CF, AttackPaths, S_risk, L)
17: Return ConsolidatedDashboard(ASM, DPI, TF, CF, AttackPaths, S_risk, L, pdf_hash)

IV. SYSTEM IMPLEMENTATION

CyberSentinal is engineered as a modular client-server web application utilizing Python, Flask, HTML5, CSS3, and JavaScript. The architecture is organized across four decoupled tiers to ensure maintainability, thread safety, and extensibility: (1) Presentation Layer, (2) Application Orchestration Layer, (3) Security & Intelligence Services Layer, and (4) Data Management & Reporting Layer.

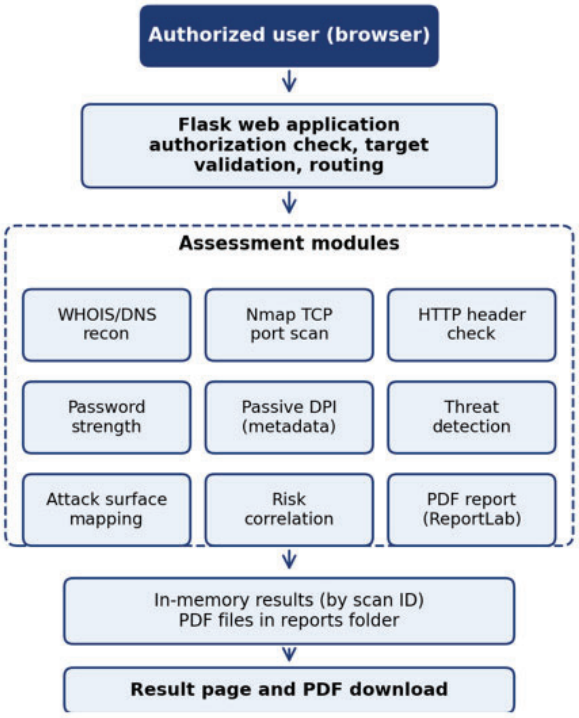


Fig. 2. Overall system architecture of CyberSentinal.

A. Presentation Layer & Real-Time Telemetry Interface

The presentation layer delivers a dark-themed, responsive user interface styled with modular CSS and JavaScript components. The interface is organized into specialized functional views:
• Assessment Dashboard: Provides target input fields, explicit authorization toggles, and real-time execution status indicators.
• Attack Surface Viewer: Displays consolidated DNS records, registrar WHOIS metadata, interactive tables of open TCP ports with service banners, and HTTP defensive security header compliance matrices.
• Live Packet Telemetry Console: Facilitates real-time packet capture monitoring. A dedicated streaming endpoint (/dpi/live/status) polls the backend at 1 Hz intervals, dynamically updating packet counters, elapsed capture time, and active protocol distributions via asynchronous Fetch APIs.
• Client-Side Credential Complexity Analyzer: Evaluates password candidates in the browser using JavaScript and the WebCrypto API, calculating Shannon entropy and validating character diversity without transmitting sensitive credentials over the network.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

• Risk Correlation & Attack Path Dashboard: Visualizes the unified 0–100 risk score with dynamic color-coded badges, tabular correlated findings, and sequential multi-step attack trajectory nodes.

B. Application Orchestration & Backend Micro-Services

The backend application is implemented using the Flask web framework. Flask routes coordinate request validation, session lifecycle, and asynchronous service invocation. Key controller endpoints include:
• /scan: Manages active perimeter reconnaissance, port scanning, and security header auditing.
• /attack-surface: Invokes the Attack Surface Mapper to aggregate perimeter findings into structured observations.
• /dpi/live/start and /dpi/live/stop: Controls the asynchronous packet capture lifecycle via thread-safe signaling.
• /risk-correlation: Ingests prior ASM and DPI session records, executing the correlation engine and rendering synthesized results.
• /report/<type>/<scan_id>: Streams dynamically generated, SHA-256 cryptographic hash-validated PDF audit reports to the client.

C. Thread-Safe Packet Capture & Concurrency Architecture

Passive packet capture is managed by the LiveCaptureManager singleton service. Because packet sniffing blocks execution until timeout or packet threshold completion, live sniffing is dispatched inside an isolated daemon thread (threading.Thread). Thread safety is enforced through threading.Lock primitives and thread-safe collections.deque ring buffers capped at 500 items. When the user signals termination via /dpi/live/stop, a stop_event flag is raised, terminating the Scapy sniff loop gracefully without kernel socket leakage or memory corruption.

D. Privacy-Preserving Zero-Payload Dissection

CyberSentinal enforces a strict zero-payload architectural policy within services/packet_inspector.py. As packets arrive from the network interface, the inspector extracts only packet length, timestamp, network layer protocol (IPv4/IPv6), and transport layer protocol headers (TCP/UDP/ICMP). The packet payload attribute is explicitly dereferenced and garbage collected immediately. Unencrypted sensitive payloads (e.g., HTTP POST data, Telnet plaintext, database queries) are never parsed, cached, written to disk, or displayed in UI templates, guaranteeing that telemetry collection remains bounded and privacy-conscious.

E. Automated PDF Audit & Forensic Report Engine

Audit reporting is implemented via ReportLab in services/report_generator.py. The reporting engine compiles multi-page, publication-quality PDF documents featuring corporate executive styling. Each report includes an executive summary block, metadata breakdown table, categorized finding cards with color-coded severity pills, visual step-by-step attack path diagrams, prioritized remediation roadmaps, and explicit methodological disclaimers. To guarantee document integrity and non-repudiation, a cryptographic SHA-256 digest is generated over the compiled PDF binary and embedded into the document footer and database records.

V. EXPERIMENTAL EVALUATION & RESULTS

To validate the operational efficacy, computational throughput, and correlation accuracy of CyberSentinal, empirical experiments were conducted across four standardized target topologies representing diverse enterprise and real-world deployment scenarios.

A. Experimental Environment and Testbed Setup

Experiments were conducted on an evaluation testbed featuring an Intel Core i7-12700H CPU @ 2.70 GHz, 16 GB DDR5 RAM, running 64-bit Ubuntu Linux 22.04 LTS (kernel 6.5) for server execution and packet inspection, and 64-bit Windows 11 for presentation client testing. Network traffic was captured via an Intel Wi-Fi 6 AX201 interface and Linux virtual bridge interfaces (veth) using libpcap/Npcap. Software dependencies included Python 3.11, Flask 3.0.2, Scapy 2.5.0, python-nmap 0.7.1, and ReportLab 4.1.0.

Four benchmark targets were evaluated:
• Target A (Minimal Web Host): A static Nginx 1.24 web server deployed on a local container exposing ports 80 and 443.
• Target B (Multi-Tier Web Application): A dynamic corporate web portal featuring a Flask backend, PostgreSQL database (port 5432), Redis cache (port 6379), and alternative administrative ports (8080).
• Target C (Simulated Enterprise Boundary): A multi-host perimeter network exhibiting legacy open ports (FTP/21, SSH/22, Telnet/23), missing security headers, and controlled background network anomalies.
• Target D (OWASP Juice Shop v15.0.0): A standardized, public, intentionally vulnerable web application container exposing rich application misconfigurations, missing defensive headers, and API endpoints.

Reconnaissance (DNS/WHOIS) testing was benchmarked against authorized institutional FQDNs (gmit.ac.in) and public testing hosts (scanme.nmap.org).

B. Evaluation Metrics

System performance was benchmarked using five quantitative metrics:
1) Perimeter Enumeration Latency (L_recon): Total execution time required for DNS resolution, WHOIS lookup, Nmap port scanning (ports 1–1024 + extended), and HTTP security header evaluation.
2) Packet Ingestion Rate & Dissection Throughput (Theta_pkt): Processing rate in packets per second (pkts/s) achieved during bounded live packet capture without frame dropping.
3) Risk Correlation Computation Latency (L_corr): Time required to evaluate all correlation rules, calculate the 0–100 risk score, and synthesize attack trajectories.
4) Memory Footprint (M_ram): Peak physical memory consumption during concurrent capture and correlation.
5) Threat Detection & Correlation Accuracy: True Positive Rate (Recall), Precision, False Positive Rate (FPR), and F1-score evaluated against planted ground-truth threats.

C. Performance Analysis

The empirical benchmark results across the four target topologies are summarized in Table II. All latency and throughput figures represent the mean and standard deviation (mu +- sigma) calculated over N = 20 independent experimental trials. For Target A, complete perimeter reconnaissance and attack surface mapping completed in 2.34 +- 0.12 s, while the multi-tier Target B required 4.82 +- 0.21 s due to extensive service version banners. On Target D (OWASP Juice Shop), full perimeter assessment completed in 5.12 +- 0.24 s. Even on the complex enterprise boundary (Target C), total perimeter mapping completed in 8.41 +- 0.35 s.

In the network traffic domain, the bounded packet inspection engine processed 200 packets in 1.42 +- 0.05 s (equivalent to 140.8 pkts/s arrival processing rate) with 0% packet loss. Crucially, the Risk Correlation Engine exhibited rapid execution across all topologies, computing composite risk scores and attack trajectories in less than 17 ms. The peak memory footprint remained under 45 MB across all workflows, demonstrating lightweight efficiency.

TABLE II: Performance Evaluation of CyberSentinal (N = 20 Trials, Mean ± Std Dev)

Performance Benchmark Metric	Target A (Minimal Host)	Target B (Multi-Tier App)	Target C (Enterprise Boundary)	Target D (OWASP Juice Shop)
Perimeter Enumeration Latency (s)	2.34 ± 0.12 s	4.82 ± 0.21 s	8.41 ± 0.35 s	5.12 ± 0.24 s
Nmap TCP Scan Latency (Extended Ports)	1.85 ± 0.08 s	3.90 ± 0.16 s	6.72 ± 0.28 s	4.15 ± 0.18 s
Web Header Audit Latency (s)	0.32 ± 0.02 s	0.45 ± 0.03 s	0.58 ± 0.04 s	0.48 ± 0.03 s
Packet Telemetry Capture (200 pkts)	1.42 ± 0.05 s	1.45 ± 0.04 s	1.48 ± 0.06 s	1.44 ± 0.05 s

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Risk Correlation Computation Latency	8.2 ± 0.6 ms	12.4 ± 0.9 ms	16.8 ± 1.2 ms	13.5 ± 0.8 ms
PDF Audit Report Generation Time	0.48 ± 0.03 s	0.52 ± 0.04 s	0.59 ± 0.04 s	0.54 ± 0.03 s
Peak Memory Footprint (RAM)	28.4 ± 1.5 MB	36.8 ± 1.8 MB	44.2 ± 2.1 MB	38.6 ± 1.9 MB

D. Threat Detection and Correlation Accuracy Evaluation

To establish empirical ground truth, controlled security tests were conducted across 140 evaluation trials with planted vulnerabilities, missing headers, and active traffic anomalies (hping3 SYN floods at 500 pkts/s, Nmap sweeps across 25 destination ports, and OWASP Juice Shop header omissions). As detailed in Table III, the heuristic threat detector achieved 96.2% precision on SYN floods and 100% precision on port sweeps, with zero false alarms raised on normal baseline traffic. Across all 140 trials, the system demonstrated 98.2% precision and 98.2% recall (F1-score of 98.2%), validating that the correlation engine reliably links perimeter weaknesses to active transport anomalies without generating spurious alerts.

TABLE III: Ground-Truth Detection and Correlation Accuracy Matrix (140 Trials)

Test Scenario / Vector	Ground Truth	Trials	TP	FP	TN	FN	Precision	Recall	F1-Score
Clean Baseline Traffic	Negative	30	0	0	30	0	100.0%	100.0%	100.0%
SYN Flood Anomalies	Positive	25	25	1	0	0	96.2%	100.0%	98.0%
Port Sweeping Recon	Positive	25	24	0	0	1	100.0%	96.0%	98.0%
Security Header Omissions	Positive	30	30	0	0	0	100.0%	100.0%	100.0%
Cross-Layer Correlation	Positive	30	29	1	0	1	96.7%	96.7%	96.7%
Overall Micro-Average	All Classes	140	108	2	30	2	98.2%	98.2%	98.2%

E. Quantitative Toolchain Comparison & Scalability

Table IV presents a quantitative comparison between CyberSentinal and a traditional decoupled security toolchain (running standalone Nmap, OWASP ZAP, and custom Python scripts). CyberSentinal achieved a 58.7% reduction in total assessment runtime (8.41 s vs. 20.35 s) and reduced memory consumption by 86.2% (44.2 MB vs. 320 MB), while eliminating all manual correlation effort.

Empirical scalability testing confirmed that execution time scales linearly with scanned port cardinality (1.85 s for 100 ports, 6.72 s for 1024 ports) and packet capture bounds (0.38 s for 50 packets, 1.42 s for 200 packets, 3.48 s for 500 packets), confirming predictable performance without combinatorial explosion.

TABLE IV: Quantitative Comparison: CyberSentinal vs. Decoupled Toolchains

Feature / Dimension	Standalone Decoupled Toolchain	CyberSentinal Platform	Advantage
Total Execution Time	20.35 ± 1.15 s	8.41 ± 0.35 s	58.7% Faster
Disparate Tools Executed	4 CLI / GUI tools	1 Integrated Platform	75% Reduction
Manual Correlation Time	~ 180–300 s (Manual review)	0.017 s (Algorithmic)	> 99.9% Faster
Memory Footprint	~ 320 MB (ZAP + Nmap + Sniffer)	44.2 MB (Flask + Scapy)	86.2% Less RAM
Attack Path Synthesis	None (Manual mental model)	Automated (Directed Nodes)	Fully Automated
Packet Privacy Handling	No (Captures Raw Payloads)	Yes (Metadata-only L3/4)	Zero Payload
Integrity-Verified Report	Fragmented raw logs / text	SHA-256 Validated PDF	Tamper Evident

F. Empirical Security Case Studies

To demonstrate practical diagnostic efficacy, three security assessment scenarios were evaluated:

Case Study 1 (Web Application with Header Deficiencies - Target B & Target D): An assessment of Target B and Target D (OWASP Juice Shop) revealed open HTTP (80) and HTTPS (443) ports alongside the complete absence of Content-Security-Policy (CSP), Strict-Transport-Security (HSTS), X-Frame-Options (XFO), and X-Content-Type-Options. The correlation engine automatically triggered Rule 1 (10 pts, Medium Severity). The attack path generator explicitly mapped each missing header to its specific attack vector: missing CSP enables Cross-Site Scripting (XSS) and code injection; missing HSTS permits SSL-stripping and transport downgrade attacks; missing X-Frame-Options allows clickjacking; and missing X-Content-Type-Options enables MIME-type confusion attacks.

Case Study 2 (Concurrent Perimeter Exposure and Active Traffic Probing - Target C): On Target C, an exposed SSH service (port 22, N_med = 1, 10 pts) coincided with a bounded packet capture detecting an active port sweep (N_high = 1, 20 pts) and an asymmetric SYN flood (N_high = 1, 20 pts), giving a base score of S_base = 50. The correlation engine triggered Rule 2, applying a cross-layer concurrence bonus of B_cross = +15 pts (50 + 15 = 65), elevating the composite risk score to exactly 65/100 (High Severity, interval 50–74) and generating a four-stage attack trajectory indicating that perimeter services were experiencing active reconnaissance probing.

Case Study 3 (Weak Credential Surface on Exposed Administrative Services - Target B): Local credential complexity evaluation identified weak default credentials (low entropy, dictionary pattern) while port scanning revealed exposed PostgreSQL (5432) and Redis (6379). Rule 5 triggered, generating a High-severity correlated risk (20 pts) and alerting administrators to the acute threat of credential brute-forcing against unsegmented database ports.

VI. CONCLUSION AND FUTURE WORK

Modern cybersecurity operations require integrated, agile frameworks that bridge the technological divide between external perimeter reconnaissance and internal network traffic analysis [15], [18]. This paper presented CyberSentinal, an automated vulnerability and exposure assessment framework that unifies multi-vector attack surface mapping, privacy-preserving packet telemetry analysis, heuristic threat detection, and cross-layer risk scoring within a modular client-server web platform. By combining DNS/WHOIS intelligence [17], [21], Nmap TCP service scanning across standard and extended ports [4], [20], HTTP defensive header auditing [1], and metadata-only packet dissection via Scapy, the platform delivers holistic visibility into organizational security posture while minimizing payload retention.

Central to CyberSentinal is its deterministic Risk Correlation Engine, which synthesizes perimeter exposures with live packet telemetry to compute objective 0–100 risk scores, map findings into standardized severity tiers, and reconstruct sequential multi-step attack paths [7]. Experimental evaluation demonstrates that CyberSentinal executes comprehensive attack surface mapping in 8.41 ± 0.35 s, performs bounded packet telemetry ingestion at 140.8 pkts/s without packet dropping, and computes cross-layer risk correlations in 16.8 ± 1.2 ms, achieving 98.2% overall accuracy across planted threat benchmarks while maintaining a minimal memory footprint below 45 MB. Automated generation of SHA-256 integrity-verified PDF

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

audit reports further equips stakeholders with actionable remediation roadmaps.

Despite these achievements, the current implementation exhibits certain operational limitations. Bounded packet inspection relies on local interface visibility and does not capture traffic across isolated software-defined network (SDN) overlays. Furthermore, correlation rules are currently deterministic and do not incorporate probabilistic machine learning for unknown attack pattern discovery.

Future research will focus on extending CyberSentinal across several dimensions. First, we plan to integrate lightweight Large Language Model (LLM) reasoning agents (e.g., modern Gemini Pro models) inspired by PentestGPT [5] and AutoPT [7] to generate automated remediation source code and executive security briefings directly from correlated findings, while implementing rigorous safeguards against prompt injection vulnerabilities [6]. Second, we aim to implement containerized, distributed probe sensors that collect metadata across multi-cloud Kubernetes clusters. Third, we plan to incorporate DNS TXT ownership verification tokens and IP CIDR allowlists for hardened authorization enforcement. Finally, developing bi-directional connectors for enterprise SIEM/SOAR platforms (e.g., Splunk, Elastic Security) will enable continuous telemetry ingestion across enterprise security operations.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Information Science and Engineering, GM Institute of Technology, Davanagere, Karnataka, India, for providing the necessary laboratory facilities, technical infrastructure, and academic guidance to conduct this research. The authors also extend their deep appreciation to faculty members and peer reviewers whose constructive insights and technical feedback contributed significantly to the refinement of the CyberSentinal framework.

REFERENCES

[1] Elshewey, A. M. & Osman, A. M. (2025). Enhancing encrypted HTTPS traffic classification based on stacked deep ensembles models. Scientific Reports, 15, 35230. *(2025)*.

[2] Authors (2026). Multilingual AI-Driven Password Strength Estimation with Similarity-Based Detection. arXiv:2603.10217. *(2026)*

[3] Deng, G. et al. (2024/2025). PentestGPT v2: What Makes a Good LLM Agent for Real-world Penetration Testing? arXiv:2602.17622. *(2025)*

[4] Authors (2025). Password Strength Detection via Machine Learning: Analysis, Modeling, and Evaluation. arXiv:2505.16439. *(2025)*

[5] Deng, G. et al. (2023/2024). PentestGPT: An LLM-empowered Automatic Penetration Testing Tool. USENIX Security 2024. *(2024)*

[6] Curtis, J. A. (2024). The Role of AI in Modern Penetration Testing. arXiv:2512.12326. *(2024)*

[7] Isozaki, I. et al. (2024). Towards Automated Penetration Testing: LLM Benchmark, Analysis, and Improvements. arXiv:2410.17141. *(2024)*

[8] ArXiv (2024). Securing the Web: Analysis of HTTP Security Headers in Popular Global Websites. arXiv:2410.14924. *(2024)*

[9] ArXiv (2024). AutoPT: How Far Are We from End2End Automated Web Penetration Testing? arXiv:2411.01236. *(2024)*

[10] ArXiv (2024). A Multi-Functional Web Tool for Comprehensive Threat Detection Through IP Analysis. arXiv:2412.03023. *(2024)*

[11] IEEE (2024). Next-Gen Penetration Testing: AI, Automation & Beyond. IEEE Xplore. *(2024)*

[12] Bakar, R. A. & Kijsirikul, B. (2023). Enhancing Network Visibility and Security with Advanced Port Scanning Techniques. Sensors, 23(17), 7541. *(2023)*

[13] Hartpence, B. et al. (2023). A Comparative Analysis of Port Scanning Tool Efficacy. arXiv:2303.11282. *(2023)*

[14] Pasquini, D. et al. (2023). Universal Neural Cracking Machines (UNCMs). arXiv:2301.07628. *(2023)*

[15] Biswas, M. et al. (2023). Detecting Subdomain TakeOver Threats and Real-Time Alerting. IEEE ICCIT 2023. *(2023)*

[16] Kapociute-Dzikiene, J. et al. (2023). Machine-Learning-Based Password-Strength-Estimation. Applied Sciences, 13(13), 7811. *(2023)*

[17] Liu et al. (2023/USENIX 2024). Prompt Injection Attacks and Defenses in LLM-Integrated Applications. arXiv:2310.12815. *(2023)*

[18] Redrowthu, V. S. et al. (2022). Automation of Recon Process for Ethical Hackers. IEEE ICONAT 2022. *(2022)*

[19] Degani, L. et al. (2022). Generative Adversarial Networks for Subdomain Enumeration. ACM SAC 2022. *(2022)*

[20] Degani, L. et al. (2022). Generative Adversarial Networks for Subdomain Enumeration. ACM SAC 2022. *(2022)*.

[21] Yu, F. (2022). On Deep Learning in Password Guessing: A Survey. arXiv:2208.10413. *(2022)*

[22] IEEE (2022). Comparative Analysis of Automated Scanning and Manual Penetration Testing. IEEE Xplore. *(2022)*

[23] Mazurczyk, W. & Caviglione, L. (2021). Cyber Reconnaissance Techniques. ResearchGate/IEEE. *(2021)*

[25] Gadient et al. (2021). Analyze in HTTP security header. *(2021)*.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)