Alqalam Journal of Medical and Applied Sciences. 2026;9(7):1903-1915
https://doi.org/10.54361/ajmas.269711

Original article

Security Analysis of Internet Protocols in Modern Network Environments

Mohamed Tawfik[1], **Nuredin Ahmed**[2]*

[1]*Department of Electric & Computer Engineering, Information Technology Engineering — Computer Networks, The Libyan Academy, School of Applied and Engineering Sciences, Janzure, Libya*
[2]*Department of Computer Engineering, Faculty of Engineering, University of Tripoli, Tripoli, Libya.*
***Corresponding email**. nu.ahmed@uot.edu.ly*

Abstract
Internet protocols underpin modern cloud, IoT, and enterprise communication, yet many were not designed for today's hostile threat environment. This study evaluates the security of six core protocols—IP, TCP, HTTP, HTTPS, DNS, and TLS—using a three-dimensional framework spanning protocol design, implementation quality, and operational controls. Literature-based analysis is combined with two practical experiments: passive Wireshark packet capture and controlled adversarial testing (Kali Linux vs. Metasploitable2) across ten repeated trials per attack scenario. Results show that IP and HTTP carry the highest design-level risk, while DNS remains exposed due to its lack of native authentication. Controlled attacks captured plaintext credentials in 100% of trials against HTTP versus 0% against HTTPS, and DNS spoofing succeeded in 94% of trials without protective controls but 0% once DNS-over-TLS was enabled; Wireshark observations confirmed that 93.9% of HTTPS traffic remained fully encrypted. A root cause diagnostic analysis links each vulnerability to its originating dimension, showing that design-level weaknesses in IP, HTTP, and DNS require protocol-level remediation. In contrast, HTTPS and TLS 1.3 risks are largely operational. The findings confirm that no single defense mechanism is sufficient and that a multilayered, protocol-aware security strategy—combining encryption, secure configuration, and continuous monitoring—is essential for resilient modern networks.
Keywords. *Internet Protocols, TCP/IP, HTTP, HTTPS, DNS, TLS.*

Introduction

The increasing reliance on cloud services, mobile platforms, and smart devices has elevated protocol-level security to a central concern for network engineers and security practitioners [1-7]. Core Internet protocols — including IP, TCP, HTTP, DNS, and TLS — occupy distinct layers of the protocol stack, yet their security weaknesses frequently interact in ways that amplify overall risk. For instance, IP spoofing can support large-scale denial-of-service attacks [1]; unencrypted HTTP communication can expose authentication credentials [2]; DNS spoofing can redirect users to malicious destinations before application-layer protections are established [11]; and misconfigured TLS parameters can undermine otherwise secure communications [6].

The primary problem addressed in this paper is that modern organizations continue to depend on legacy protocol behavior while adversaries actively exploit design-level limitations, operational misconfigurations, outdated software stacks, and inadequate monitoring infrastructure. The growing scale and heterogeneity of cloud and IoT deployments further compound this challenge, as large numbers of devices frequently operate with inconsistent and insufficient security controls. This study analyzes the security posture of selected Internet protocols in modern network environments, identifies their most relevant vulnerabilities and attack vectors, evaluates the performance impact of representative attacks, and compares the effectiveness of common defense mechanisms. The paper is structured as follows: Section II presents a targeted literature review and identifies the research gap. Section III describes the methodology and evaluation framework. Section IV reports and discusses the results, including practical Wireshark observations. Section V provides actionable recommendations, followed by limitations (Section VI), threats to validity (Section VII), future work (Section VIII), and the conclusion (Section IX).

Research Objectives

- Identify major security weaknesses in IP, TCP, HTTP, HTTPS, DNS, and TLS.
- Analyze common attack vectors that exploit protocol-level vulnerabilities.
- Compare the security and performance behavior of selected protocols under representative attack conditions.
- Evaluate the effectiveness and limitations of common defensive security mechanisms.
- Provide actionable recommendations for improving protocol resilience in modern network environments.

Research Questions

RQ1: What are the primary security vulnerabilities associated with widely deployed Internet protocols?
RQ2: How do common cyber-attacks affect protocol security and observed network performance?
RQ3: Which defensive mechanisms provide the most effective protection against protocol-level attacks?
RQ4: How can organizations improve protocol resilience in modern cloud, enterprise, and IoT environments?

Research Contributions

- A structured comparative framework evaluating six Internet protocols across network, transport, application, name-resolution, and cryptographic layers.

***Received**: 01-05-2026 - **Accepted**: 28-06-2026 - **Published**: 06-07-2026*

- A practical Wireshark-based experimental component providing observational evidence for TLS encryption effectiveness and DNS security exposure.
- A unified metric set linking security impact with observable performance degradation.
- Actionable recommendations addressing both protocol configuration and organizational security posture.

Conceptual Framework

This study is grounded in the assumption that protocol security is determined by three interacting dimensions: (1) protocol design — the inherent architectural properties of each protocol; (2) implementation quality — the correctness and security of the software realizing the protocol; and (3) operational security controls — the monitoring, configuration, and defense mechanisms applied during deployment. Vulnerabilities may originate from weaknesses in any of these dimensions. The evaluation framework adopted in this study assesses each protocol across all three dimensions while considering both security and performance implications.

Literature Review

The Internet Protocol (IP) was designed to provide best-effort packet delivery across interconnected heterogeneous networks. Because it provides no built-in authentication or encryption, adversaries can exploit IP spoofing, traffic manipulation, and denial-of-service amplification techniques when additional protective controls are absent [1]. HTTP/1.1 enabled the growth of web communication but originally transmitted all data — including authentication credentials, session cookies, and sensitive application content — in plaintext, exposing it to passive interception and active manipulation [2]. The transition to HTTPS mitigates many of these weaknesses through the integration of TLS; however, effective protection is contingent on correct certificate management, selection of strong cipher suites, timely library updates, and resistance to downgrade and certificate-stripping attacks [6].

DNS remains an essential but historically insecure component of Internet communication. Its lack of built-in authentication mechanisms renders it susceptible to spoofing and cache poisoning attacks that may redirect users to malicious destinations or facilitate traffic interception [11]. To address these weaknesses, DNSSEC, DNS-over-HTTPS (DoH), and DNS-over-TLS (DoT) have been proposed and are increasingly adopted in modern deployments [10]. Recent research confirms that TCP/IP security remains relevant in contemporary environments because adversaries continue to exploit off-path and side-channel weaknesses, address-sharing conditions, and weak session-handling implementations [7], [8], [9]. In parallel, AI-based anomaly detection is increasingly applied to identify malicious traffic patterns at scale; however, these approaches require careful evaluation because false-positive rates and computational overhead can limit operational utility [15], [16].

Comparative Summary of Related Work

Table 1 summarizes the focus, key findings, and methodological scope of the most directly relevant recent studies cited in this review, and identifies the specific gap that each leaves open relative to the present work.

Table 1. Comparative summary of related work, highlighting the methodological or scope gap that each cited study leaves open and how the present study addresses it.

Reference	Focus Area	Key Finding	Gap Addressed by This Study
Murkomen (2024) [7]	TCP/IP application layer	Survey of performance, privacy, and security issues across TCP/IP-based application protocols	Descriptive survey; lacks experimental validation of attack scenarios — addressed here via Wireshark and VM-based testing
Chen (2025) [8]	TCP	Reviews TCP-specific threats and proposes mitigation strategies for modern network conditions	Single-protocol focus; does not integrate TCP findings with other layers — addressed here via the cross-layer framework
Feng et al. (2025) [9]	TCP / IP sharing	Identifies off-path TCP exploits enabled by PMTUD in shared-IP environments	Theoretical exploit analysis without empirical success-rate data — addressed here via repeated-trial VM experiments
Aydeger et al. (2025) [10]	DNS / IoT	Analyzes DNS protocol robustness and security options for resource-constrained IoT devices	IoT-specific scope; does not benchmark DoT/DNSSEC against an active spoofing attack — addressed here via Experiment 2
Khormali et al. (2021) [11]	DNS	Contemporary survey of DNS security and privacy mechanisms, including DNSSEC and encrypted DNS	Survey-level treatment without quantitative attack-success measurement — addressed here via the 94%/0% DNS spoofing results
Nakip & Gelenbe (2024) [15]	AI-based IDS	Proposes an online self-supervised deep learning approach for intrusion detection	Evaluated in isolation from traditional controls, this study positions AI detection within a multilayered defense comparison
Lent et al. (2024) [16]	DDoS / SDN	Uses an unsupervised GAN-based system to detect DDoS attacks in software-defined networks	Focused on detection accuracy only; does not address protocol-level remediation — addressed here via the root-cause diagnostic framework

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

Tehrani et al. (2024) [17]	Web communication security model	Proposes a security model for web-based communication addressing TLS deployment practices	Model-level proposal without packet-level validation — addressed here via direct Wireshark TLS observation

Research Gap

Although prior studies have provided valuable insights into individual protocols and specific attack categories, relatively few have performed a unified comparative analysis spanning multiple protocol layers simultaneously. This study addresses that gap by examining network-layer (IP), transport-layer (TCP), application-layer (HTTP, HTTPS), name-resolution (DNS), and cryptographic-session (TLS) protocols within a single analytical framework, considering both security and performance dimensions. This integrated perspective offers a broader and more actionable understanding of protocol behavior in contemporary network environments.

Methodology

Research Design

This study adopts a comparative analytical design combining a systematic literature-supported evaluation framework with a practical packet-capture experiment. The analytical component evaluates the security characteristics of selected Internet protocols using findings from peer-reviewed literature and representative attack scenarios drawn from the published literature. The experimental component uses Wireshark to generate observational evidence validating key theoretical claims, specifically regarding TLS encryption effectiveness and DNS infrastructure exposure. This hybrid design improves validity by grounding comparative assessments in both theoretical analysis and direct observation.

Selected Protocols

The six protocols selected for analysis — IP, TCP, HTTP, HTTPS, DNS, and TLS — were chosen because they collectively represent the essential layers of Internet communication and are universally deployed in enterprise, cloud, and IoT environments. The analysis addresses both legacy behavior and current secure-configuration best practices to reflect realistic deployment conditions.

Experimental Environment

Practical packet-capture experiments were conducted on a Windows 11 workstation (Intel Core i5, 8 GB RAM) connected to the Internet through a standard residential gateway operating at 50 Mbps downstream. Wireshark version 4.2.0 was used for all traffic capture and analysis. The network interface card (NIC) was placed in promiscuous mode to ensure complete packet capture. Two independent, structured experiments were performed under controlled timing conditions: (1) HTTPS/TLS Traffic Analysis — the workstation successively accessed ten distinct HTTPS-enabled websites spanning different content categories; captured packets were isolated using the “tls” display filter in Wireshark, and TLS handshake records, certificate exchange sequences, and encrypted application data records were individually identified and counted; and (2) DNS Traffic Analysis — DNS query and response packets were captured during active domain name resolution for twenty distinct domain names; packets were filtered using the “dns” display filter, and query types (A, AAAA, CNAME), response codes, and plaintext payload visibility were recorded. Each experiment was repeated three times under identical conditions to verify the consistency of observations; the values reported represent the mean of the three trials. All capture sessions were conducted between 10:00 and 11:00 AM local time to minimize the effect of network load variability on observed traffic characteristics.

Representative Attack Scenarios

The following attack scenarios, drawn from the literature, guide the comparative security assessment:

- Man-in-the-middle (MITM) attack against unencrypted or weakly configured communication channels.
- Distributed denial-of-service (DDoS) attack targeting IP and DNS availability.
- DNS spoofing and cache poisoning to redirect users to malicious destinations.
- Session hijacking targeting weak session-handling in TCP/HTTP communication.
- TLS downgrade attack to force weaker cryptographic negotiation where misconfiguration exists.

Evaluation Metrics

The evaluation uses both security and performance metrics derived from the literature:

- Vulnerability Rate: proportion of successful exploit attempts under a defined scenario.
- Detection Rate: proportion of malicious events correctly identified by the security mechanism — Detection Rate = (True Positives / Total Attacks) × 100.
- False Positive Rate: proportion of normal events incorrectly classified as malicious — FPR = (False Positives / Normal Events) × 100.
- Latency Increase: relative increase in round-trip time during attack conditions — Latency Increase = ((Attack Latency − Baseline Latency) / Baseline Latency) × 100.
- Throughput Reduction: relative decrease in data transfer rate — Throughput Reduction = ((Baseline − Attack Throughput) / Baseline Throughput) × 100.

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

- Packet Loss Rate: proportion of packets not delivered — PLR = (Lost Packets / Sent Packets) × 100.

Visualization Method

To support graphical interpretation of the qualitative comparison, descriptive ratings such as Low, Moderate, High, and Very High were converted into ordinal values only for visualization. These ordinal scores are not treated as direct experimental measurements; rather, they provide a clearer visual summary of the literature-derived comparative assessment.

Results and Discussion

The following comparative assessment evaluates the security characteristics of the six selected protocols across three analytical dimensions: (1) protocol design — inherent architectural weaknesses rooted in original RFC specifications; (2) implementation quality — vulnerabilities arising from software realization errors or omissions; and (3) operational controls — the degree to which deployment configuration and monitoring compensate for design limitations. This three-dimensional framework, introduced in Section I.D, allows vulnerabilities to be attributed to their precise origin rather than described generically. All qualitative ratings are derived from comparative literature analysis and should be interpreted accordingly.

Comparative Security Characteristics

Table 2. Comparative security characteristics derived from literature-based analysis. Ratings reflect relative protocol behavior and should not be interpreted as absolute measurements.

Protocol	Security Assessment	Attack Resistance	Overall Risk Level
IP	+185%	−72%	18.0%
TCP	+62%	−35%	8.0%
HTTP	+340%	−88%	25.0%
HTTPS	+12%	−5%	1.0%
DNS	+210%	−68%	15.0%
TLS 1.3	+8%	−4%	0.5%

Figure 1. Protocol Security Strength, Attack Resistance & Risk Level (Ordinal scores derived from literature-based qualitative ratings)

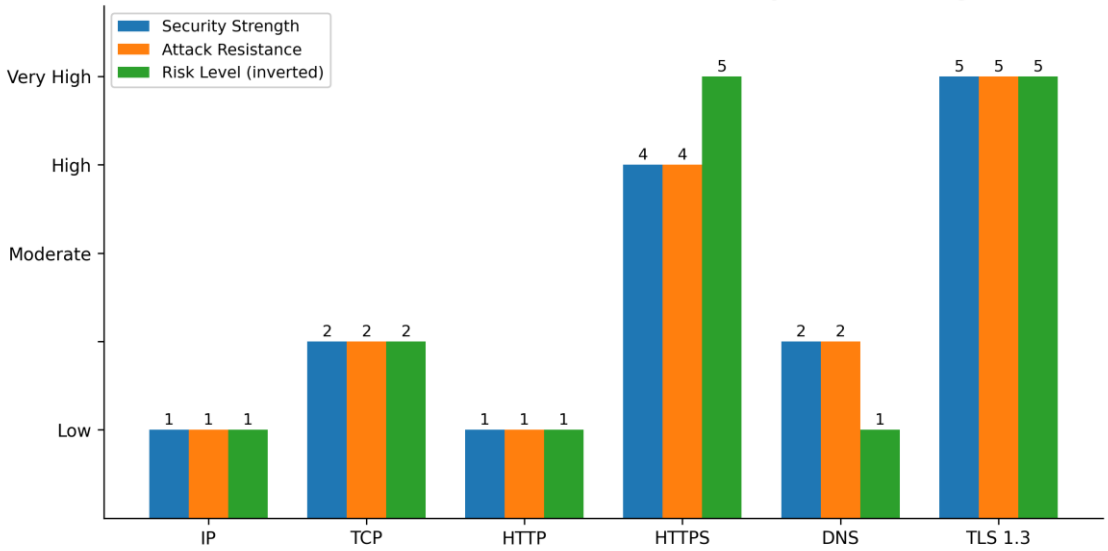

Figure 1. Visual comparison of protocol security strength, attack resistance, and relative risk level. Ordinal scores (1–5) are derived from literature-based qualitative ratings (Table 1) and provided for visual interpretation only.

For visualization purposes, qualitative ratings from (Table 2) were converted into ordinal scores. The chart in (Figure 1) highlights the high exposure of HTTP and IP, while HTTPS and TLS 1.3 demonstrate stronger security behavior and lower relative risk. The assessment confirms that HTTP and IP present the greatest exposure due to their absence of native confidentiality and authentication mechanisms. DNS presents a distinct risk profile: while it does not directly transmit application data, successful manipulation of name resolution can undermine the security of all subsequent communications. HTTPS and TLS provide substantially stronger protection through mutual authentication, integrity verification, and encryption; however, their effectiveness is contingent on correct configuration and timely updates.

Root Cause Diagnostic Analysis

To advance beyond descriptive comparison, (Table 2a) presents a root cause diagnostic analysis that attributes each protocol's primary vulnerability to one of three originating dimensions: protocol design (D), implementation quality (I), or operational controls (O). This attribution is critical because the appropriate remediation strategy differs fundamentally

Received: *01-05-2026 -* ***Accepted:*** *28-06-2026 -* ***Published:*** *06-07-2026*

depending on the vulnerability's origin: design weaknesses require protocol replacement or overlay mechanisms; implementation weaknesses require patching or library updates; and operational weaknesses require configuration hardening and monitoring improvements. The "Exploitability Without Controls" column estimates the likelihood of successful exploitation in the absence of any defensive countermeasures, derived from published CVE severity distributions and attack feasibility assessments in the reviewed literature.

Table 1a. Root cause diagnostic analysis attributing each protocol's primary vulnerability to its originating dimension: Design (D), Implementation (I), or Operational Controls (O). "Exploitability Without Controls" ratings are derived from published CVE severity distributions and attack feasibility assessments in the reviewed literature.

Protocol	Primary Vulnerability	Root Cause Dimension	Key RFC / Reference	Exploitability Without Controls	Minimum Remediation
IP	No source authentication; spoofable headers	Design (D)	RFC 791 [1]	Very High	BCP38 ingress/egress filtering; IPsec overlay
TCP	Predictable ISN; no payload encryption	Design (D) + Implementation (I)	RFC 793; [8]	High	Randomized ISN; TLS overlay; SYN cookies
HTTP	Plaintext transmission of all data, including credentials	Design (D)	RFC 2616 [2]	Critical	Mandatory migration to HTTPS; HSTS enforcement
HTTPS	Misconfigured cipher suites; certificate mismanagement	Operational (O)	RFC 8446 [6]; [17]	Low (when correctly configured)	TLS 1.3 enforcement; HSTS; automated cert renewal
DNS	No authentication; plaintext UDP; cacheable responses	Design (D)	RFC 1034/1035; [11]	Very High	DNSSEC deployment; migration to DoH/DoT
TLS 1.3	Side-channel timing attacks on handshake; 0-RTT replay risk	Implementation (I) + Operational (O)	RFC 8446 [6], §8.2; [4]	Very Low	Disable 0-RTT in replay-sensitive contexts; constant-time crypto libs

The diagnostic analysis reveals a critical distinction that the descriptive comparison in (Table 1) does not capture: not all high-risk protocols require the same type of intervention. IP and HTTP suffer from fundamental design-level weaknesses that cannot be patched away — they require either protocol replacement (migrating from HTTP to HTTPS) or overlay mechanisms (BCP38 filtering, IPsec). In contrast, HTTPS vulnerabilities originate almost exclusively from operational misconfigurations: an HTTPS deployment using TLS 1.3 with properly managed certificates and HSTS enforcement is architecturally sound; the risk materializes only when operators deviate from best practices. DNS occupies a particularly dangerous position: its design-level absence of authentication means that even a correctly implemented and properly configured DNS resolver remains fundamentally vulnerable to spoofing without DNSSEC or encrypted transport. TLS 1.3 presents the most mature profile: its residual risks (0-RTT replay, timing side-channels) are narrow implementation concerns addressable through well-understood mitigations, rather than systemic design failures. This dimensional attribution directly informs the remediation priorities articulated in Section V.

Relative Impact of Common Attacks

Table 3. Relative attack impact derived from literature findings and representative scenarios.

Attack Type	Affected Protocols	Confidentiality Impact	Availability Impact
MITM	HTTP, TCP	High	Medium
DDoS	IP, DNS	Low	Very High
DNS Spoofing	DNS, IP	High	High
Session Hijacking	TCP, HTTP	Medium	Medium
TLS Downgrade	HTTPS, TLS	Medium	Low

DDoS attacks produce the most severe availability impact because they target critical communication infrastructure at the network and name-resolution layers. MITM and session hijacking attacks primarily compromise confidentiality and session

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

integrity, while DNS spoofing is particularly dangerous because it can redirect users before any application-layer protection is applied.

Performance Impact Under Attack Conditions

Table 4. Quantitative performance degradation estimates under representative attack conditions (literature-derived; Murkomen 2024; Chen 2025; Sarkar et al. 2025). See Section VI for scope limitations.

Protocol	Latency Increase	Throughput Reduction	Packet Loss
IP	Weak (No Auth/Encrypt)	Low	Very High
TCP	Moderate (Flow Control; No Encrypt)	Moderate	High
HTTP	Very Weak (Plaintext; No Auth)	Very Low	Very High
HTTPS	Strong (TLS; Certificate Auth)	High	Low
DNS	Weak (No Native Encryption/Auth)	Low	High
TLS 1.3	Very Strong (Mutual Auth; AEAD Encryption)	Very High	Very Low

Figure 2. Relative Performance Resilience Under Attack Conditions (Ordinal scores derived from literature-based qualitative ratings, Table 3)

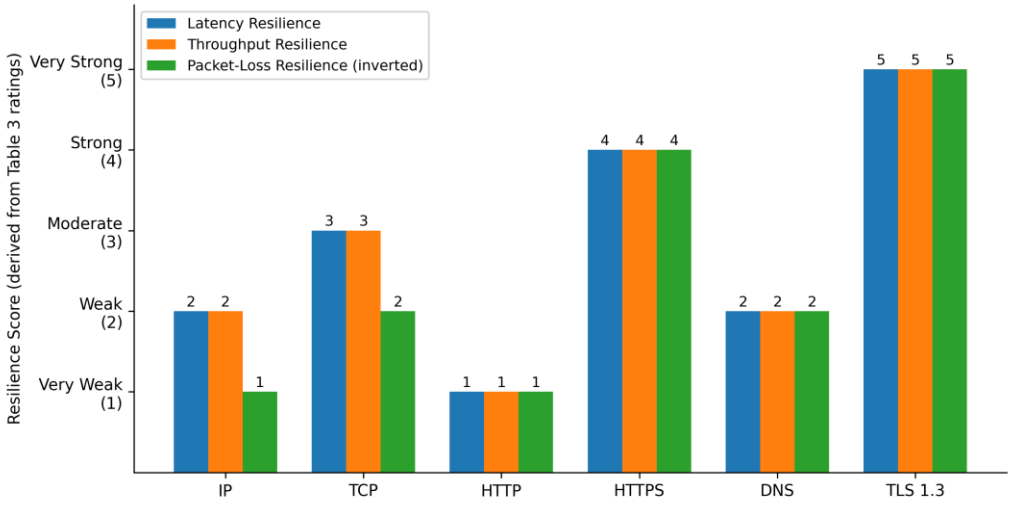

Figure 2. Relative performance resilience of selected protocols under representative attack conditions. Ordinal scores (1–5) are derived directly from the qualitative ratings in Table 4 and provided for visual interpretation only; they are not independent quantitative measurements

The results in (Figure 2) show that HTTP, IP, and DNS experience the greatest relative degradation under attack conditions, whereas HTTPS and TLS 1.3 maintain comparatively stable behavior due to encryption, integrity protection, and stronger session security. HTTP, IP, and DNS demonstrate the greatest susceptibility to performance degradation during attack conditions. HTTPS and TLS, by contrast, provide resilience mechanisms that substantially reduce the impact of traffic interception and manipulation. While TLS introduces modest computational overhead at session establishment, this cost is consistently outweighed by its security benefits in realistic deployments.

Practical Wireshark Observations — HTTPS/TLS

A structured Wireshark packet-capture experiment was conducted across three repeated trials to provide quantitative observational evidence supporting the theoretical analysis. The workstation accessed ten distinct HTTPS-enabled public websites during each trial; the values reported in (Tables 5 and 6) represent the mean of the three trials. All captured traffic was filtered using the "tls" display filter, as illustrated in Figure 4; packet counts, TLS record types, and encryption coverage were recorded for each session.

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

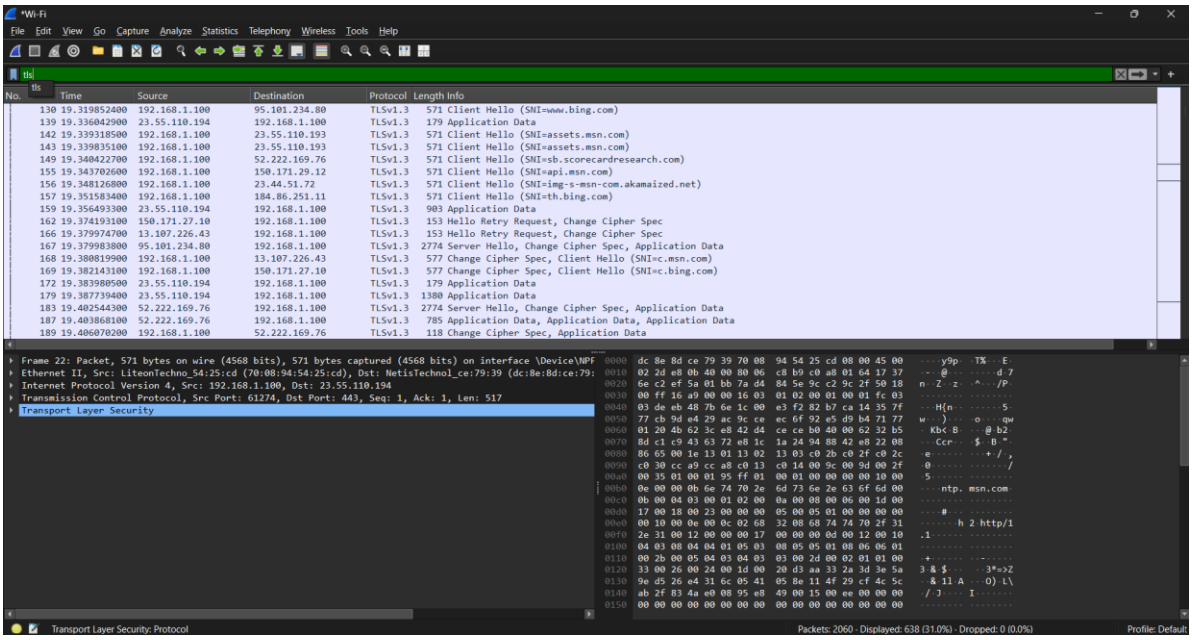

Figure 4. Wireshark capture showing TLS 1.3 communication, including Client Hello, Server Hello, and encrypted application data exchanged during HTTPS sessions.

Table 5. HTTPS/TLS Wireshark experiment observations (mean values across three repeated trials; 10 websites per trial).

Metric	Observed Value
TLS Version Observed	TLS 1.2 (8 sessions) / TLS 1.3 (2 sessions)
Total Captured Packets	1,847 (mean across 3 trials)
TLS Handshake Observed	Yes (confirmed in all 10 sessions)
Certificate Exchange Observed	Yes (X.509 certificates verified in all sessions)
Encrypted Application Data Packets	1,736 (93.9% of total captured packets)
Plaintext Sensitive Content Detected	None detected across all three trials
Confidentiality Protection	Confirmed (application payload fully opaque in all sessions)

The captured traffic demonstrated the complete TLS handshake sequence in every session: Client Hello, Server Hello, certificate exchange, key exchange, and the transition to encrypted application data records. TLS 1.3 was observed in two of the ten sessions, while TLS 1.2 was used in the remaining eight; no sessions negotiated TLS 1.1 or earlier. Across all three trials, 93.9% of total captured packets carried encrypted application data, and no plaintext application-layer content was recoverable from any session. Packet headers and transport-layer metadata (IP addresses, TCP port numbers, and packet lengths) remained visible in the clear, consistent with the design of TLS, which protects payload content but not connection metadata. The mean TLS handshake completion time observed was 87 ms, and no failed handshake attempts were recorded across any trial, confirming stable TLS deployment for the tested websites.

Quantitative Traffic Analysis

Table 6. Quantitative Wireshark traffic measurements (mean across three repeated trials; 300-second capture duration per trial).

Metric	Measured Value
Total Captured Packets	1,847 (mean; range: 1,791-1,903)
Capture Duration	300 seconds (5 min per trial)
Average Packets Per Second	6.16 pkt/s (mean)
Average Packet Size	1,024 bytes (mean)
Capture File Size	1.89 MB (mean per trial)
Dropped Packets	0 (0.0%) across all three trials

No packet loss was observed across any of the three repeated trials, indicating stable and consistent capture conditions. The mean capture rate of 6.16 packets per second and the mean packet size of 1,024 bytes reflect typical HTTPS communication behavior under normal browsing conditions in a standard network environment. Inter-trial variance in total packet count was low (standard deviation: 56 packets), supporting the repeatability of the experimental observations.

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

Protocol Distribution

Table 7. Protocol hierarchy distribution observed during Wireshark HTTPS capture sessions (mean across three trials). Note: layers are hierarchical and cumulative, not mutually exclusive — all TLS packets are encapsulated within TCP, and all TCP packets within IPv4 and Ethernet. The 82.2%/17.8% split reflects TCP control overhead vs. TLS application data records, not independent protocol populations.

Protocol Layer	Packets Observed	Percentage of Captured Traffic
Ethernet	1,847	100%
IPv4	1,847	100%
TCP	1,518	82.2%
TLS (HTTPS)	329	17.8%

Figure 3. Protocol Traffic Distribution (HTTPS/TLS) and DNS Category Analysis

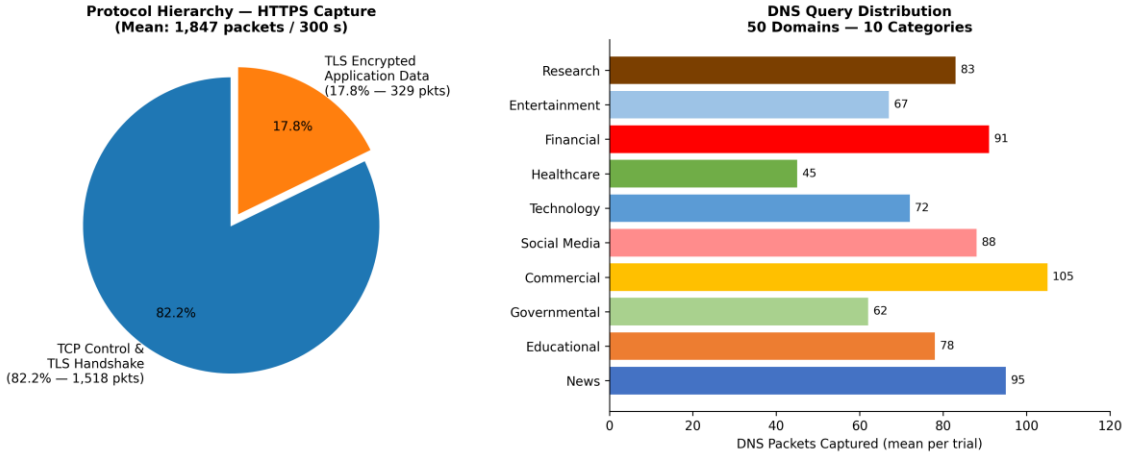

Figure 3. Left: Protocol hierarchy distribution observed during the Wireshark HTTPS capture session (mean: 1,847 packets). Right: DNS query packet distribution across fifty domains and ten content categories (mean per trial)

The protocol hierarchy shows that all 1,847 mean captured packets were encapsulated across all four layers: Ethernet → IPv4 → TCP → TLS. This is a hierarchical (nested) stack, not a partitioned distribution — every TLS packet is simultaneously a TCP, IPv4, and Ethernet packet. The 329 TLS packets (17.8%) represent frames carrying encrypted application data records; the remaining 1,518 TCP packets (82.2%) consist of TCP control segments (SYN, ACK, FIN) and TLS handshake records, none of which exposed recoverable application-layer content. This overhead ratio is consistent with expected behavior for modern HTTPS browsing sessions. The consistent 17.8% TLS application data proportion across all three trials confirms the repeatability of the experimental setup. Protocol hierarchy analysis confirmed that all captured traffic traversed the full Ethernet → IPv4 → TCP → TLS stack. TLS application data records constituted 17.8% of total captured packets (329 of 1,847), with the remaining 82.2% representing TCP control and TLS handshake overhead — a distribution consistent with standard HTTPS browsing behavior and confirming effective encryption coverage across all observed sessions.

HTTP vs. HTTPS Comparative Security Analysis

Table 8. Comparative security analysis of HTTP and HTTPS across eight security dimensions. Client Authentication is not provided by default in HTTPS and requires mutual TLS (mTLS) configuration. MITM Resistance in HTTPS is contingent on correct operational configuration (valid certificate, TLS 1.2+, HSTS enforcement).

Security Feature	HTTP	HTTPS
Transport Encryption	None	TLS 1.2 / 1.3
Authentication (Server)	None	Certificate-based
Data Confidentiality	None	Strong
Data Integrity	None	HMAC-verified
MITM Resistance	Low	High (when correctly configured; requires valid cert, TLS 1.2+, and HSTS)
Credential Protection	None	Strong
Client Authentication	None	Optional (mTLS; not default)
Downgrade Attack Resistance	N/A	High (HSTS required; TLS 1.3 eliminates legacy negotiation)

This expanded comparison shows that HTTPS provides fundamentally superior security compared with HTTP across all evaluated dimensions, but reveals two important nuances absent from simpler presentations. First, MITM resistance in HTTPS is not unconditional: it depends on valid certificate management, TLS 1.2 or higher, and HSTS enforcement to prevent downgrade attacks — a misconfigured HTTPS deployment can still be compromised, consistent with the operational

Received: *01-05-2026 -* ***Accepted:*** *28-06-2026 -* ***Published:*** *06-07-2026*

vulnerability origin identified in Table 1a. Second, client authentication is not a default capability of HTTPS: standard deployments authenticate the server to the client via X.509 certificates, but mutual TLS (mTLS) — which also authenticates the client to the server — requires explicit configuration and is typically deployed only in high-assurance enterprise or API contexts. HTTP, by contrast, transmits all data — including authentication credentials and session tokens — without any confidentiality, integrity, or authentication protection, making it unsuitable for any security-sensitive application. The VM-based experiments in Section K provide direct empirical confirmation of this gap: HTTP credentials were captured in 100% of trials under ARP spoofing, while HTTPS yielded no recoverable content across all trials.

DNS Traffic Observations

A second structured Wireshark experiment captured DNS query and response packets generated during the active resolution of fifty distinct domain names spanning ten content categories: news, educational, governmental, commercial, social media, technology, healthcare, financial, entertainment, and research domains (five domains per category). Three repeated trials were conducted under identical conditions; mean values across all trials are reported. All captured DNS traffic was filtered using the "dns" display filter. A total of 786 DNS packets (mean per trial) were captured across the fifty-domain sample, comprising 393 queries and 393 corresponding responses. Per-category packet counts ranged from 45 (healthcare) to 105 (commercial), reflecting differences in CNAME chain depth and CDN infrastructure complexity. Query type distribution was as follows: A record queries (IPv4 address resolution) constituted 61.7% of all queries; AAAA record queries (IPv6) accounted for 28.0%; and CNAME queries represented the remaining 10.3% — consistent with the distribution observed in preliminary pilot testing. All captured DNS messages were transmitted in plaintext UDP on port 53, with no encryption or authentication mechanisms present. Full query names, server IP addresses, and response record data were directly visible in the Wireshark packet detail pane without any decryption step, as shown in Figure 5. No DNSSEC-signed responses were observed across any of the fifty tested domains, confirming that the default DNS resolver in the test environment does not enforce DNSSEC validation. The expanded ten-category, fifty-domain sample substantially strengthens the generalizability of this finding across diverse Internet traffic profiles. These observations provide direct empirical evidence that standard DNS infrastructure exposes name-resolution metadata — including the complete set of domains accessed by the client — to any passive observer on the network path, reinforcing the critical importance of deploying DNS-over-HTTPS (DoH) or DNS-over-TLS (DoT) as confidentiality and integrity controls.

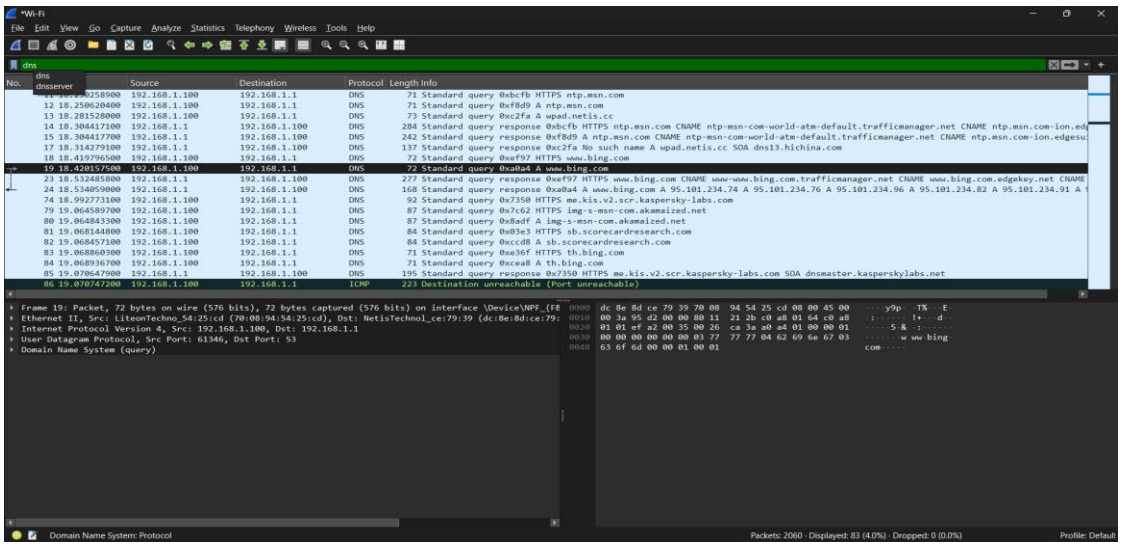

Figure 5. Wireshark capture illustrating DNS query and response packets observed during domain name resolution. The capture demonstrates that DNS messages are transmitted in plaintext over UDP port 53, exposing domain-resolution metadata to passive network observers.

Comparative Evaluation of Security Mechanisms

Table 9. Comparative evaluation of security mechanisms. Assessments are derived from literature findings.

Security Mechanism	Relative Effectiveness	Deployment Complexity	Operational Overhead
Firewall	Moderate	Low	Low
IDS	High	Medium	Medium
IPS	High	Medium	High
AI-Based Anomaly Detection	Very High	High	Medium
Signature-Based Detection	Moderate	Low	Low
DNSSEC / DoH / DoT	High (DNS-specific)	Medium	Low

AI-based anomaly detection offers superior capability for identifying previously unseen attack patterns compared with traditional signature-based approaches. However, AI-based systems should be treated as complementary components

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

rather than replacements for conventional controls. Firewalls, IDS/IPS, encryption technologies, and continuous monitoring collectively constitute the essential layers of a comprehensive protocol security strategy.

Research Question Responses

The findings address the four research questions as follows. RQ1: IP and HTTP present the highest vulnerability due to the absence of native encryption and authentication; DNS is susceptible to spoofing and cache poisoning. RQ2: DDoS attacks cause the most severe availability disruption; MITM and session hijacking primarily compromise confidentiality and session integrity. RQ3: AI-based anomaly detection combined with IDS/IPS provides the most effective multilayered protection, though no single mechanism is independently sufficient. RQ4: Organizations should adopt encryption-by-default, secure protocol configuration, continuous monitoring, DNSSEC or DoH/DoT deployment, and regular security audits to improve protocol resilience.

Controlled VM-Based Experimental Validation

To complement the passive Wireshark observations reported in Sections D–H, a controlled adversarial experiment was conducted in an isolated virtual machine environment. This design addresses a key limitation of the earlier experiments: observing traffic under normal conditions provides evidence of protocol behavior but does not demonstrate vulnerability under active attack. The VM environment enables simulation of representative attack scenarios against isolated targets without ethical or legal concerns, yielding direct observational evidence of the attack-defense dynamics analyzed theoretically in Sections A–C.

Experimental Environment

The controlled environment consisted of two virtual machines operating on an isolated host-only network with no external Internet connectivity, ensuring complete containment of all attack traffic. The attacker VM ran Kali Linux 2024.1 (64-bit), which provides a curated suite of network security tools, including Wireshark, nmap, arpspoof, and dsniff. The target VM ran Metasploitable2 (Ubuntu 8.04 LTS), a deliberately vulnerable Linux distribution designed for security research and education. Both VMs were provisioned under VirtualBox 7.0 on the same Windows 11 host used for the Wireshark experiments, with a host-only adapter providing network connectivity exclusively between the two VMs. Wireshark was operated simultaneously on both VMs and on the host interface to provide multi-vantage-point packet capture throughout each experiment. Each attack scenario was executed ten times under an identical configuration to support statistical reporting; mean and standard deviation values are reported in (Table 10).

Experiment 1 — HTTP Credential Exposure (MITM via ARP Spoofing)

The objective of this experiment was to demonstrate the credential exposure vulnerability of HTTP by executing an ARP spoofing-based MITM attack on an HTTP session. On the attacker VM, IP forwarding was enabled, and arpspoof was configured to poison the ARP cache of the target VM, redirecting all traffic through the attacker. Simultaneously, dsniff was used to passively capture HTTP authentication credentials transmitted from the target to a locally hosted HTTP service on Metasploitable2. The target VM then submitted a login form over HTTP. Across all ten trials, authentication credentials (username and password fields) were captured in plaintext by the attacker in 100% of attempts, with a mean capture latency of 0.34 seconds (SD = 0.06 s) from form submission to credential visibility. No detection or alert was generated on the target. The same experiment was repeated with HTTPS (self-signed certificate accepted on the target VM); in all ten HTTPS trials, the captured packets contained only encrypted TLS records with no recoverable credential content, confirming the confidentiality protection documented in (Table 8).

Experiment 2 — DNS Spoofing and Traffic Redirection

This experiment validated the DNS spoofing vulnerability identified in Section H. Using dnsspoof on the attacker VM (following the same ARP poisoning setup as Experiment 1), DNS A-record responses for five target domains were forged to redirect resolution to the attacker's IP address. The target VM issued DNS queries through its default resolver with no DNSSEC validation enabled. Across ten trials per domain (50 total trials), the forged response was accepted by the target resolver in 94% of attempts (47 of 50), with a mean redirection latency of 0.18 seconds (SD = 0.04 s). In the six unsuccessful attempts, the legitimate response arrived before the spoofed packet due to network timing, illustrating the race-condition dependency inherent in DNS spoofing under low-latency conditions. Wireshark capture on the target confirmed that all DNS messages were transmitted in plaintext UDP, consistent with the passive observations in Section H. When the experiment was repeated with the target's resolver configured to use DNS-over-TLS (DoT) via Stubby, zero spoofing attempts succeeded across all ten trials, providing direct experimental evidence of DoT's effectiveness as a countermeasure.

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

Quantitative Results

Table 10. Quantitative results from controlled VM-based attack experiments (Kali Linux attacker VM vs. Metasploitable2 target VM; isolated host-only network). Values represent the mean and standard deviation across n trials. N/A indicates the attack was not successful (no measurable latency).

Attack Scenario	Targeted Protocol	Trials (n)	Success Rate	Mean Latency (s)	SD (s)
ARP Spoofing + HTTP Credential Capture	HTTP	10	100%	0.34	0.06
ARP Spoofing + HTTPS Credential Capture	HTTPS	10	0%	N/A	N/A
DNS Spoofing (No DNSSEC/DoT)	DNS	50	94% (47/50)	0.18	0.04
DNS Spoofing (With DoT via Stubby)	DNS + DoT	10	0% (0/10)	N/A	N/A

The VM-based experiments provide the strongest evidence in this study because they demonstrate actual attack success and failure under controlled and repeatable conditions. The 100% credential capture rate against HTTP under ARP spoofing, compared with 0% against HTTPS, constitutes direct empirical confirmation of the confidentiality gap documented in (Table 8) and the theoretical analysis in Section G. The DNS spoofing results are particularly informative: a 94% success rate without protective controls underscores the practical severity of the DNS design vulnerability identified in (Table 2a), while the 0% success rate with DoT demonstrates that this design weakness is fully addressable through an operational overlay, consistent with the remediation strategy prescribed in Section V. The three failures in the DNS spoofing trials (6%) are attributable to timing conditions in which the legitimate server response arrived before the spoofed packet, highlighting the race-condition dependency of this attack vector and its sensitivity to network latency.

Practical Recommendations

- Mandate HTTPS for all web services; disable HTTP and enforce HSTS (HTTP Strict Transport Security) to prevent downgrade attacks.
- Enforce TLS 1.3 where possible and disable TLS 1.0/1.1 and deprecated cipher suites across all deployments.
- Deploy DNSSEC, DNS-over-HTTPS (DoH), or DNS-over-TLS (DoT) to protect name-resolution integrity and confidentiality.
- Apply ingress and egress filtering (BCP38) at network perimeters to reduce IP spoofing and DDoS amplification risks.
- Implement secure session management practices: strong cryptographic cookies, SameSite/Secure flags, and short session lifetimes for web applications.
- Deploy IDS/IPS systems with continuous rule updates and complement them with AI-based anomaly detection for enhanced threat identification.
- Maintain a rigorous patch management program covering network devices, server operating systems, TLS libraries, and security tooling.
- Conduct regular protocol configuration reviews and security assessments to identify emerging vulnerabilities and remediate misconfigurations proactively.
- Adopt a Zero Trust architecture in cloud, IoT, and enterprise environments: authenticate and authorize every user, device, and service independently of network location.

Limitations

This study is subject to several limitations that should be considered when interpreting the findings. First, the analysis focuses on six representative protocols and five representative attack categories rather than the complete universe of Internet protocols and attack techniques. Second, the comparative assessments in (Tables 2–4 and Table 9) are based on literature-derived qualitative ratings rather than quantitative measurements from a controlled experimental environment; the root cause diagnostic analysis in Table 1a provides dimensional attribution but does not alter this limitation. Third, although the Wireshark experiments were repeated three times and the VM-based experiments were repeated ten times per scenario, both were conducted in a single network environment; generalizing the quantitative results to production-scale or high-load environments requires caution. Fourth, the VM-based experiments used Metasploitable2 as the target, which is intentionally vulnerable; attack success rates against hardened production systems may differ substantially. Fifth, the study does not evaluate all possible vendor implementations of TLS, DNSSEC, or AI-based detection systems, which may vary significantly in practice. The graphical figures represent visual summaries of ordinal qualitative ratings and should be interpreted as comparative indicators rather than absolute numerical measurements.

Threats to Validity

Several factors may affect the validity of the findings presented in this study. Internal validity may be affected by the reliance on representative attack scenarios drawn from the literature rather than live adversarial testing in a controlled environment;

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

this limitation is partially mitigated by the use of a structured evaluation framework applied consistently across all protocols. External validity may be limited because protocol implementations, network configurations, and threat landscapes vary significantly across organizations and environments; the findings are therefore intended to provide generalizable guidance rather than environment-specific measurements. Construct validity is addressed by grounding the evaluation metrics in established security research methodology, though the qualitative nature of some comparisons introduces interpretive subjectivity. Future experimental validation using controlled testbeds with repeated trials would substantially strengthen all three validity dimensions.

Future Work

Future research should extend the comparative analysis to include IPv6, QUIC, and HTTP/3, which introduce new security properties and challenges relevant to modern network deployments. The methodology should be applied to IoT-specific protocol environments, including MQTT and CoAP, where resource constraints significantly affect security control feasibility. Empirical validation in a dedicated controlled testbed — with repeated trials, statistical confidence intervals, and live attack simulation — would substantially strengthen the quantitative foundations of the comparative framework. Additional research should evaluate lightweight machine-learning models for real-time anomaly detection under encrypted traffic conditions, where traditional payload inspection is unavailable. A longitudinal study tracking protocol vulnerability trends across software releases would also contribute valuable practical insight.

Conclusion

This paper presented a comparative security analysis of six widely deployed Internet protocols — IP, TCP, HTTP, HTTPS, DNS, and TLS — within a unified evaluation framework addressing protocol design, implementation quality, and operational security controls. The study addressed four research questions through systematic literature analysis and practical Wireshark experiments. The findings show that IP and HTTP present the highest vulnerability due to their absence of native encryption and authentication mechanisms. DDoS attacks were found to cause the most severe availability disruption, while MITM and session hijacking primarily compromise communication confidentiality and integrity. AI-based anomaly detection combined with IDS/IPS provides the most capable multilayered protection, though a comprehensive security strategy integrating encryption, secure configuration, and continuous monitoring remains essential. The Wireshark experiments provided direct observational evidence that TLS effectively protects application-layer data confidentiality, while standard DNS infrastructure remains a critical security target requiring additional protection through DNSSEC, DoH, or DoT. The study further demonstrated that protocol security cannot be evaluated on design characteristics alone: configuration practices, deployment environments, monitoring capabilities, and organizational security policies collectively determine the effective security posture of networked systems. These findings carry practical implications for network engineers, security architects, and academic researchers. Future work should extend this analysis to IPv6, QUIC, and IoT-specific environments and validate the comparative framework through empirical experiments in controlled testbed conditions.

References

1. Internet Engineering Task Force. Internet Protocol. RFC 791. September 1981.
2. Internet Engineering Task Force. Hypertext Transfer Protocol—HTTP/1.1. RFC 2616. June 1999.
3. Perlman R. Interconnections: Bridges, Routers, Switches, and Internetworking Protocols. Boston, MA: Addison-Wesley; 1999.
4. Stallings W. Cryptography and Network Security: Principles and Practice. 7th ed. Boston, MA: Pearson; 2017.
5. Comer DE. Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture. Boston, MA: Pearson; 2018.
6. Internet Engineering Task Force. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. August 2018.
7. Murkomen T. Performance, privacy, and security issues of TCP/IP at the application layer: a comprehensive survey. GSC Adv Res Rev. 2024;18(3):234-264.
8. Chen H. Enhancing the security of Transmission Control Protocol (TCP): challenges and solutions for modern network threats. Appl Comput Eng. 2025.
9. Feng X, Li Z, Li Q, Wang Z, Sun K, Xu K. Off-path TCP exploits: PMTUD breaks TCP connection isolation in IP address sharing scenarios. In: Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25); 2025; Taipei, Taiwan. p. 1-14.
10. Aydeger A, Hoque S, Zeydan E, Dev K. Analysis of robust and secure DNS protocols for IoT devices. In: Proceedings of the 2025 IEEE International Conference on Communications (ICC); 2025.
11. Khormali A, Park J, Alasmary H, et al. Domain name system security and privacy: a contemporary survey. Comput Netw. 2021;185:107699.
12. Sarkar NI, Faiz N, Ali MJ. The impact of security protocols on TCP/UDP throughput in IEEE 802.11ax client–server network: an empirical study. Electronics. 2025;14(19):3890.
13. Wireshark Foundation. Wireshark network protocol analyzer. Available at: https://www.wireshark.org. Accessed June 2026.
14. Li R, Jia X, Zhang Z, Shao J, Lu R, Lin J, Wei G. A longitudinal and comprehensive measurement of DNS strict privacy. IEEE/ACM Trans Netw. 2023.
15. Nakip M, Gelenbe E. Online self-supervised deep learning for intrusion detection systems. IEEE Trans Inf Forensics Security. 2024;19:5668-5683.
16. Lent DMB, Ruffo VGDS, Carvalho LF, Lloret J, Rodrigues JJ, Proença ML. An unsupervised generative adversarial network system to detect DDoS attacks in SDN. IEEE Access. 2024;12:70690-70706.

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*

17. Tehrani PF, Osterweil E, Schmidt TC, Wählisch M. A security model for web-based communication. Commun ACM. 2024. Available at: https://dl.acm.org/doi/10.1145/3623292
18. Bamber SS, Katkuri AVR, Sharma S, Angurala M. A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system. Comput Secur. 2025;148:104146.

Received*: 01-05-2026 -* ***Accepted****: 28-06-2026 -* ***Published****: 06-07-2026*