

A Comprehensive Review of Phishing-Resistant Web Authentication and Continuous Session Security: Mechanisms, Threat Landscape, and Emerging Frameworks

Aryan Parmar¹, Dr. Rachna Patel²

¹Dept. of Computer Science & Engineering, C.G.P.I.T., Uka Tarsadia University, Bardoli, Gujarat, India

Abstract - Modern web applications rely on authentication mechanisms that verify user identity exclusively at login, leaving active sessions exposed to a broad range of post-authentication threats. As enterprises increasingly adopt distributed, API-first architectures, the consequences of session-level compromise have grown substantially — a single stolen token can grant persistent access to sensitive data, administrative functions, and interconnected microservices. This paper presents a comprehensive review of the evolution of web authentication technologies — from password-based systems and token-based mechanisms such as OAuth 2.0 and JSON Web Tokens (JWT), to phishing-resistant protocols including WebAuthn and FIDO2 — with a focused examination of the structural gap in continuous session security. Existing authentication standards provide strong initial validation but lack provisions for monitoring session integrity after login, creating an exploitable window for token theft, session hijacking, and account takeover attacks. This review systematically analyses the session threat taxonomy, formal adversary models, and the limitations of current standards as identified by NIST SP 800-63B and the OWASP API Security framework. Furthermore, emerging approaches to continuous authentication — including behavioral baseline profiling, IP/device fingerprint consistency checks, request velocity analysis, risk-based re-authentication, and zero-trust session enforcement — are reviewed and compared across deployment feasibility, false positive rates, detection latency, and computational overhead. Findings indicate that lightweight, software-only continuous validation frameworks represent the most deployable path forward, requiring no machine learning dependencies or hardware modifications while achieving sub-5-second mean detection latency and less than 5 ms per-request overhead. The review identifies key open challenges including threshold optimization for heterogeneous user populations, browser fingerprinting evasion by sophisticated adversaries, mobile WebAuthn integration complexity, and the absence of standardized continuous session interfaces, providing a structured research agenda for future work in session-aware authentication systems.

Key Words: Web authentication, phishing-resistant authentication, session hijacking, continuous authentication, JWT, OAuth 2.0, WebAuthn, FIDO2, zero-trust architecture, session security, behavioral baseline, risk scoring

1. INTRODUCTION

Authentication is the foundational mechanism by which web systems establish the identity of users seeking access to protected resources. From the earliest networked systems relying on shared secrets to today's cryptographic challenge-response protocols, authentication has been the primary line of defence against unauthorized access. As web applications have grown in complexity — encompassing multi-tier microservice architectures, third-party integrations, mobile clients, and IoT endpoints — the attack surface surrounding identity verification has expanded correspondingly. Yet a fundamental architectural limitation has persisted across this entire evolution: authentication is treated as a one-time event at login. Once a credential such as a JSON Web Token (JWT) or session cookie is issued and validated, it typically grants unrestricted access for a fixed lifetime — commonly 15 to 60 minutes — regardless of any subsequent changes in the context, environment, or behavior of the session [1].

This structural limitation creates a broad and underappreciated attack window. Adversaries who successfully intercept a valid token after login can achieve complete account compromise without triggering any re-authentication prompt. Such attacks do not require breaking cryptographic primitives — they exploit the implicit trust that systems place in authenticated sessions after initial verification [2]. The attacker need not steal the user's password, defeat multi-factor authentication, or bypass a phishing-resistant login ceremony; they simply need to obtain the token that was issued following these protections. Once in possession of a valid session token, the adversary inherits all privileges of the legitimate user for the token's remaining lifetime, with no mechanism in any current major authentication standard to detect or interrupt this impersonation [3].

The threat landscape surrounding active session exploitation has grown considerably more sophisticated in recent years. Credential stuffing campaigns have expanded their scope to target active sessions via cross-site scripting (XSS) attacks, network interception on untrusted Wi-Fi networks, and malware-based token exfiltration from browser storage. OAuth

2.0 misconfigurations expose long-lived refresh tokens that can be silently replayed long after the legitimate user has closed their browser. JWT "alg:none" and algorithm confusion vulnerabilities persist in production deployments of popular libraries despite being publicly documented for nearly a decade [3]. Man-in-the-browser attacks inject malicious scripts into authenticated sessions without modifying the URL or triggering standard security controls.

The attack surface spans three distinct temporal phases of the authentication lifecycle. The pre-login phase — encompassing phishing, credential stuffing, brute force enumeration, and SIM swapping — has been substantially addressed by WebAuthn and FIDO2 specifications, which provide cryptographic origin binding that defeats phishing and eliminates reusable credentials [4]. The login moment is addressed by adaptive MFA systems and risk-based authentication engines deployed by major identity providers [5]. However, the active session phase remains largely unaddressed by any major authentication standard, creating what this paper terms the post-authentication security gap [6].

A standards gap analysis confirms this deficiency across the authoritative specifications governing web identity. OAuth 2.0 RFCs mandate token validation but leave session monitoring entirely unspecified [7]. JWT RFC 7519 requires signature verification without any behavioral context checks [8]. WebAuthn provides phishing-resistant login ceremonies but explicitly excludes session management from its specification scope [4]. FIDO2 CTAP operates only during registration and initial authentication phases [9]. NIST SP 800-63B acknowledges risk-based re-authentication as a valuable concept but provides no concrete implementation guidance [10].

This paper reviews the state of the art in web authentication security with particular focus on the session security gap and emerging solutions. Section 2 reviews the evolution of web authentication mechanisms from passwords through modern phishing-resistant protocols. Section 3 examines the session threat landscape, providing a detailed taxonomy and formal adversary model. Section 4 surveys continuous authentication research. Section 5 presents a structured comparative analysis. Section 6 discusses open challenges and a future research agenda. Section 7 concludes the paper.

2. WEB AUTHENTICATION: EVOLUTION AND STANDARDS

2.1 Password-Based Systems and Their Limitations

Password-based authentication has been the dominant mechanism for web identity verification since the earliest networked systems. Bonneau et al. [1] established the canonical security-usability-deployability (SUD) framework for evaluating password replacement alternatives, confirming that no single mechanism outperforms passwords across all three dimensions simultaneously. Passwords score well on deployability but score poorly on security, being vulnerable to phishing, credential stuffing, offline dictionary attacks, and credential reuse across services.

The security limitations of static passwords have motivated a succession of compensating controls. Salted hashing using algorithms such as bcrypt, scrypt, and Argon2 has substantially raised the computational cost of offline cracking. However, hashing provides no protection against phishing or credential reuse attacks. Bursztein et al. [11] demonstrated through large-scale empirical analysis that password reuse is widespread and systematic, enabling credential stuffing attacks that achieve non-trivial success rates even against services with strong hashing.

Multi-factor authentication (MFA) was introduced to compensate for inherent password weaknesses by requiring a second verification factor. The most widely deployed second factors include TOTP codes, SMS-based OTPs, and push notifications. However, TOTP codes and SMS OTPs are themselves phishable through adversary-in-the-middle proxy attacks. SMS-based OTP delivery is additionally vulnerable to SIM swapping attacks [2]. These limitations motivated the development of phishing-resistant authentication mechanisms.

2.2 Token-Based Authentication: OAuth 2.0 and JWT

The introduction of token-based authentication architectures, particularly OAuth 2.0 [7] and OpenID Connect [12], addressed the problem of credential delegation and federated identity in multi-party web ecosystems. Rather than sharing passwords directly with third-party applications, resource owners grant authorization to clients via an authorization server that issues access tokens — cryptographically signed credentials that convey specific permissions for a defined scope and duration.

JSON Web Tokens [8] became the predominant format for OAuth 2.0 access tokens due to their self-contained nature. A JWT encodes claims along with an expiry timestamp and a cryptographic signature in a compact URL-safe format that can be validated by any party holding the signing key without consulting a central authority. However, formal security analyses have revealed structural weaknesses in these protocols. Fett et al. [3] revealed authorization bypasses arising from misconfigured redirect URIs and CSRF vulnerabilities. Mainka et al. [13] enumerated SSO token replay vulnerabilities that persist across widely deployed implementations.

A particularly critical limitation of JWT-based sessions is their revocability. Because JWTs are validated stateless-ly, a server cannot revoke a token before its expiry without implementing a token denylist — introducing stateful dependencies that undermine the scalability advantages of stateless JWT validation. This means that a stolen JWT remains fully valid for its entire remaining lifetime following theft, which may be 15 to 60 minutes in typical configurations [6].

2.3 Phishing-Resistant Authentication: WebAuthn and FIDO2

The FIDO Alliance's WebAuthn specification [4], standardized by the W3C as the Web Authentication API, represents the most significant advancement in primary authentication security in recent years. WebAuthn implements public-key challenge-response authentication using hardware authenticators. During registration, the authenticator generates a cryptographic key pair bound to the relying party; the private key is stored securely within the authenticator and never transmitted.

The phishing resistance of WebAuthn derives from its cryptographic binding of authentication ceremonies to the relying party origin. A phishing site operating at a different domain cannot forge a valid WebAuthn response bound to the legitimate origin, because the authenticator will refuse to sign a challenge for an origin that does not match the registered origin [4]. Large-scale deployments at major technology companies have confirmed that WebAuthn adoption effectively eliminates phishing-based account takeover.

Despite these strong security properties, WebAuthn's specification scope is strictly limited to the authentication ceremony at login. Session management following the authentication ceremony is explicitly outside the scope of the WebAuthn specification [4]. This means that even applications implementing WebAuthn for phishing-resistant login remain fully vulnerable to post-authentication token theft and session hijacking.

2.4 Federated Identity and Single Sign-On

OpenID Connect (OIDC) [12] extended OAuth 2.0 with a standardized identity layer, enabling relying parties to verify user identity through a trusted identity provider using JWT-formatted ID tokens. OIDC enabled the widespread adoption of social login via established identity providers such as Google, Microsoft, Apple, and Facebook.

However, federated architectures introduce their own security challenges. Gajek et al. [14] formally analyzed Single Sign-On protocols, identifying vulnerabilities in token binding, session fixation, and cross-site request forgery. A particularly significant concern is the concentration of risk: a compromise of the identity provider propagates immediately to all relying parties that depend on it. Sivakorn et al. [15] demonstrated practical OAuth phishing attacks despite PKCE adoption, leveraging redirect URI validation weaknesses to intercept authorization codes through open redirectors.

2.5 Emerging Authentication Approaches

Passkeys — a consumer-friendly implementation of WebAuthn credentials that synchronize across devices via cloud keychain services — address the usability limitation of hardware security keys by enabling phishing-resistant authentication without requiring a separate hardware device [4]. Passkeys store the FIDO2 private key in the device's secure enclave and synchronize the encrypted key material to the user's cloud account.

Continuous Access Evaluation Protocol (CAEP) and Shared Signals Framework (SSF), developed under the OpenID Foundation, represent an emerging standards-track approach to post-issuance token revocation and session event propagation. CAEP defines a protocol for communicating security events between identity providers and relying parties in real time, enabling relying parties to revoke or downgrade tokens in response to events detected by the identity provider [10].

3. SESSION SECURITY: THREAT LANDSCAPE AND ADVERSARY MODELS

3.1 Session Threat Taxonomy

Post-authentication session security encompasses the protection of authenticated state after the login ceremony has succeeded and a session token has been issued. Four primary attack vectors characterize this domain, spanning different attacker capabilities, access methods, and behavioral signatures, as summarized in Table 1.

Table -1: Web Session Threat Taxonomy

| Phase | Attack Vector | Impact | HPRAF-CA Mitigation | Detection Difficulty | Signal |
|---------------------|-----------------------------------|-----------------------|-------------------------------|----------------------|------------------|
| Token Active | Network interception (MITM/XSS) | Account takeover | IP prefix validation | High | IP prefix change |
| Device Compromise | Malware token exfiltration | Full data access | Device fingerprint check | Very High | UA hash change |
| Behavioral Drift | Legitimate to attacker pivot | Silent compromise | Request rate analysis | High | Velocity spike |
| Idle Takeover | Screenlock bypass / session reuse | Unauthorized access | Idle timeout + re-auth | Medium | Idle gap |
| Token Replay | JWT replay from anomalous IP | Account impersonation | Geolocation deviation scoring | High | IP + UA mismatch |
| Refresh Token Abuse | Long-lived token exfiltration | Persistent access | Velocity + device binding | Very High | Multi-signal |

Token theft via network interception exploits the transmission of session credentials in HTTP Authorization headers on each request. While TLS encryption substantially mitigates passive network interception, XSS vulnerabilities enable JavaScript-based token extraction from browser storage, bypassing transport security entirely.

Device compromise through malware represents the most difficult attack vector to detect, as the attacker operates from the legitimate user's device. Malware-based token exfiltration silently copies session credentials from browser storage to attacker-controlled infrastructure. The transition from legitimate device usage to attacker replay from a different environment is precisely the signal that behavioral baseline monitoring can detect [6].

Behavioral drift represents the most temporally subtle attack vector. The key insight is that while the token itself remains cryptographically valid, the behavioral context of its usage has changed in ways that are statistically distinguishable from legitimate variation [6]. Idle session takeover exploits the common practice of issuing tokens with lifetimes that extend beyond typical periods of user inactivity.

3.2 Standards Gap Analysis

A systematic review of authentication standards reveals a consistent and significant gap in session security provisions across all major specifications. Applications assembling authentication infrastructure from these standards have no standardized guidance for the session security layer. Table 2 maps the major authentication standards and their coverage of session security dimensions.

Table -2: Gap Analysis of Session Security Approaches

| Reference | Login Auth | Session Monitor | Risk Scoring | Web Auth | Coverage | Evaluated | Scope |
|--------------------------|------------|-----------------|--------------|----------|---------------|-----------|-----------------|
| Bonneau et al. [1] | ✓ | — | — | — | Login only | Survey | Pre-login |
| FIDO2/WebAuthn [4] | ✓ | — | — | ✓ | Login only | Standard | Pre-login |
| OAuth 2.0 [7] | ✓ | — | — | — | Token mgmt | RFC | Pre-login |
| Fett et al. [3] | ✓ | — | — | — | Login only | Formal | Pre-login |
| Alaca & van Oorschot [5] | ✓ | Partial | — | — | Biometrics | Survey | Session partial |
| NIST SP 800-63B [10] | ✓ | Partial | Concept | — | Guideline | NIST | Session partial |
| OWASP API Top 10 [16] | ✓ | — | — | ✓ | Practitioner | OWASP | Pre-login |
| HPRAF-CA [6] | ✓ | ✓ | ✓ | ✓ | Full coverage | Empirical | Full session |

NIST SP 800-63B [10] comes closest to addressing session security, acknowledging the concept of risk-based re-authentication and recommending periodic re-verification for sensitive transactions. However, the specification leaves implementation entirely to the relying party, providing only high-level goals without the technical specificity needed to guide consistent implementation. The OWASP API Security Top 10 [16] focuses on defensive coding practices but does not address active session monitoring or anomaly detection.

3.3 Formal Adversary Model for Session Attacks

Drawing on Shostack's threat modeling methodology [17] and the formal web security models developed by Fett et al. [3], we define a session security adversary model encompassing four capability classes. A network observer adversary (A1) can observe and capture valid JWTs and replay them from attacker-controlled infrastructure with different network characteristics. A device compromise adversary (A2) controls the legitimate client following malware compromise and may operate from the legitimate device itself, making it the hardest to detect.

An automation adversary (A3) deploys high-volume automated requests using stolen session tokens, with request velocity as the primary distinguishing characteristic. An idle takeover adversary (A4) exploits session tokens during periods of user inactivity. From this adversary model, five security objectives are derived: S01 (Session Integrity), S02 (Behavioral Continuity), S03 (Minimal Friction, <2% false positives), S04 (Performance Neutrality, <10 ms overhead), and S05 (Universal Deployability — no ML or hardware required) [6].

4. CONTINUOUS AUTHENTICATION: RESEARCH LANDSCAPE

4.1 Behavioral Biometrics and Multi-Factor Continuity

Continuous authentication research emerged from recognition that the binary model of authentication — fully authenticated after login until token expiry — is insufficient for extended sessions. Alaca and van Oorschot [5] provided the first comprehensive survey of multi-factor continuous authentication approaches, cataloguing methods based on behavioral biometrics, cognitive biometrics, and physiological signals.

Keystroke dynamics analysis exploits characteristic typing timing patterns — key hold time, inter-key interval, and digraph latencies — which are sufficiently individual to distinguish users with equal error rates as low as 1-4% in controlled conditions. However, practical deployment presents significant challenges: enrollment requires hundreds of keystrokes before monitoring can begin, and keystroke dynamics provides no coverage during periods when the user is not actively typing.

Mouse dynamics-based continuous authentication tracks cursor movement velocity, acceleration, curvature, and click timing, demonstrating classification accuracy comparable to keystroke dynamics. However, mouse monitoring is inherently limited to desktop environments and provides no signal in mobile or API-client contexts. Gait-based and physiological continuous authentication approaches provide signals available even when the user is not actively interacting, but require specialized sensor hardware not universally available [5].

4.2 Network and Behavioral Baseline Profiling

An alternative paradigm for continuous session monitoring relies on network and request-level behavioral signals rather than biometric measurements. This approach requires no client-side instrumentation, is compatible with all client types, collects no physiologically sensitive data, and can be implemented as server-side middleware [6]. The core mechanism is the establishment of a behavioral baseline at session initiation and continuous comparison of incoming requests against this baseline.

IP-based geolocation validation compares the IP prefix of incoming requests against the prefix observed at session initiation. A token being used from a significantly different IP prefix is a strong signal of token theft and replay from attacker-controlled infrastructure. IP prefix matching at the /16 level provides a balance between sensitivity to geographic displacement and tolerance for expected IP address changes within the same ISP.

Device fingerprinting at the HTTP/TLS layer aggregates multiple server-observable attributes into a composite device identifier without requiring client-side JavaScript. Key signals include the User-Agent header, Accept-Language header, TLS cipher suite negotiation order, and HTTP/2 settings frame parameters. Request velocity profiling establishes a statistical baseline of the session's API request rate, flagging deviations exceeding 4x the baseline rate as indicative of automated script-based abuse [6].

4.3 Risk Scoring and Composite Signal Integration

Individual behavioral signals each provide useful information about session legitimacy, but each is subject to false positives when considered in isolation. The key insight of risk-based session monitoring is that these signals should be combined into a composite risk score, with each signal contributing weighted evidence compared against a threshold [18].

The composite risk scoring model assigns risk points to each detected anomaly: IP geolocation drift contributes 2.0 risk points; device fingerprint changes contribute 2.0 risk points; request velocity exceeding 4x the baseline contributes 1.0 risk point; idle timeout violations contribute 1.0 risk point. A composite score at or above 3.0 triggers immediate WebAuthn re-authentication. Empirical evaluation across 12,847 simulated web transactions demonstrates a false positive rate of 1.3% and an attack detection rate of 99.2% [6].

4.4 Risk-Based Re-Authentication

Risk-based re-authentication systems trigger step-up authentication when the session risk score exceeds a defined threshold, while allowing low-risk requests to proceed without interruption. WebAuthn biometric re-authentication — leveraging the device's built-in fingerprint sensor or face recognition via the FIDO2 platform authenticator API — provides a low-friction step-up mechanism consistent with the phishing-resistant posture of the initial login [4]. The ceremony can be completed in under two seconds, minimally disrupting legitimate users while providing strong cryptographic verification.

The server-side implementation responds to high-scoring requests with an HTTP 403 response containing a challenge header specifying the required re-authentication method, rather than immediately terminating the session. This design allows legitimate users who trigger the threshold to quickly re-verify their identity and continue their session without losing their work [6].

4.5 Zero-Trust Session Enforcement

The zero-trust security model, formalized in NIST SP 800-207 [19], extends the principle of least-privilege to trust itself, asserting that no request should be implicitly trusted based on its network origin or prior authentication status. In the session security context, zero-trust enforcement means that every request to a protected resource independently undergoes security validation, regardless of token age or session history.

Park and Usman [20] surveyed zero-trust architecture implementations, identifying per-request validation, continuous authorization, microsegmentation, and behavioral monitoring as core implementation components. Software-only implementations maintain session behavioral state in a distributed in-memory cache and perform stateless risk computation per request using the cached baseline. Empirical evaluation demonstrates 4.2 ms mean per-request validation overhead, well within the performance budget [6].

5. COMPARATIVE ANALYSIS OF SESSION SECURITY APPROACHES

Table 3 presents a structured comparison of the session security approaches surveyed in this review, evaluated across seven dimensions derived from the security objectives and deployment constraints identified in Section 3.

Table -3: Comparative Analysis of Session Security Mechanisms

| Approach | Phish. Resist. | Session Monitor | FP Rate | Overhead | ML Req. | HW Req. | Deploy | Notes |
|----------------------------|------------------|-----------------|-------------|---------------|-----------|----------------|-------------|------------------------|
| Password + TOTP [1] | Low | None | — | Negligible | No | No | Easy | Phishable OTP |
| OAuth 2.0 + JWT [7,8] | Medium | None | — | <1 ms | No | No | Easy | No revocation |
| WebAuthn/FIDO2 [4,9] | Very High | None | — | ~5 ms | No | Authenticator | Medium | No session coverage |
| Keystroke Biometrics [5] | N/A | Partial | 3-8% | 2-10 ms | Yes | Optional | Complex | Requires training data |
| Mouse Dynamics [5] | N/A | Partial | 4-9% | 3-15 ms | Yes | Mouse | Complex | Desktop only |
| IP/Device Baseline [6] | N/A | Full | 1.3% | 4.2 ms | No | No | Easy | Best deployability |
| Risk-Based Re-auth [10,18] | N/A | Full | 2-5% | Variable | Optional | No | Medium | Needs calibration |
| Zero-Trust [19,20] | N/A | Full | Varies | <10 ms | Optional | No | Medium | Arch. philosophy |
| Hybrid HPRAF-CA [6] | Very High | Full | 1.3% | 4.2 ms | No | Opt. HW | Easy | Best overall |

The comparison reveals several important findings. First, no single existing approach simultaneously provides phishing-resistant login AND active session monitoring. WebAuthn/FIDO2 provides the strongest initial authentication security but offers zero session monitoring; behavioral biometric approaches provide session monitoring but no phishing resistance at login, and their deployment complexity and false positive rates (3-9%) limit practical adoption.

Second, the hybrid approach — combining WebAuthn phishing-resistant login with IP/device/velocity behavioral session monitoring — achieves the best overall profile: very high phishing resistance at login, full session monitoring coverage, 1.3% false positive rate, 4.2 ms per-request overhead, no ML requirements, and straightforward server-side deployment as middleware. This hybrid approach satisfies all five security objectives defined in the adversary model [6].

Third, the performance comparison confirms that software-only, rule-based session validation is practically viable within the 10 ms per-request overhead budget. All network/request-level behavioral approaches achieve overhead well below this threshold, while ML-based approaches introduce variable overhead depending on model complexity. The rule-based approach's deterministic low overhead is a significant practical advantage for latency-sensitive API contexts.

6. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

6.1 Threshold Optimization Across Heterogeneous User Populations

The composite risk threshold must be calibrated for the specific user population and deployment context of each application. A threshold appropriate for an enterprise application serving employees from fixed office locations may generate unacceptable false positive rates for a consumer application serving internationally mobile users. Personalized threshold adaptation — calibrating the threshold for each individual user based on their historical session behavior — represents a promising research direction, though it introduces cold-start challenges for new users and security implications if a patient attacker slowly shifts the threshold toward their own behavioral profile [18].

6.2 Browser Fingerprinting Evasion and Anti-Detection

Device fingerprinting provides a useful session continuity signal under the assumption that attackers cannot perfectly replicate the victim's browser fingerprint. This assumption holds for most practical attackers but is not absolute. Browser fingerprinting evasion toolkits enable sophisticated adversaries to spoof User-Agent strings and TLS fingerprint parameters. Future research should investigate fingerprinting signals more difficult to replicate, including CPU timing side channels, hardware-based attestation signals via WebAuthn's attestation mechanism, and network timing characteristics [15].

6.3 Mobile Platform Integration and Cross-Platform Consistency

IP address volatility is inherently higher on mobile devices, which frequently transition between Wi-Fi and cellular networks. These transitions cause IP prefix changes that would trigger elevated risk scores under a framework calibrated for desktop users. Mobile-aware session monitoring may need to incorporate signals specific to the mobile context — carrier network identifiers, approximate geolocation, or accelerometer-based mobility signals — to distinguish expected mobile IP transitions from anomalous token replay [4], [9].

6.4 Machine Learning Enhancement and Adaptive Detection

The software-only, rule-based risk scoring approach demonstrates strong empirical performance with simple, interpretable rules. However, rule-based systems are inherently limited in capturing complex, nonlinear behavioral patterns. Unsupervised anomaly detection using autoencoders or isolation forests could learn a richer representation of legitimate session behavior, enabling detection of subtle anomalies that do not trigger any individual rule. Lightweight ML models — binary decision trees, logistic regression — can achieve inference times of 1-2 ms and may provide meaningful accuracy improvements while remaining within the performance budget [18].

6.5 Standardization and Interoperability

The absence of any standardized interface for continuous session validation represents a significant barrier to widespread adoption. The OpenID Foundation's SSF and CAEP represent promising steps toward standardization, but remain limited to event propagation between identity providers and relying parties rather than within-session behavioral monitoring. A proposed standard interface for per-request session risk scoring and re-authentication triggering — potentially as an extension to the OAuth 2.0 Token Introspection protocol — would enable interoperable implementations across web frameworks [10].

6.6 Privacy-Preserving Session Monitoring

Behavioral session monitoring involves the collection of request metadata that may constitute personal data under GDPR, CCPA, and similar frameworks. Privacy-preserving approaches including differential privacy techniques, federated monitoring approaches where risk computation occurs on the client device, and cryptographic commitment schemes represent important research directions. The tension between privacy and security — where more detailed behavioral data enables more accurate anomaly detection — requires careful analysis to identify designs that maintain acceptable security efficacy.

7. CONCLUSION

This paper has presented a comprehensive review of web authentication security, tracing the evolution from password-based systems through federated token architectures to phishing-resistant WebAuthn and FIDO2 specifications, and providing detailed examination of the structural session security gap that persists across all current standards. The review has established that post-authentication session monitoring represents the most critical unaddressed vulnerability in modern web authentication architecture: while the field has made substantial progress in securing the login moment, the active session remains almost entirely unmonitored by any standard mechanism.

The review of continuous authentication research reveals a clear convergence toward lightweight, server-side behavioral monitoring as the most practically deployable approach. Behavioral biometric methods provide theoretically comprehensive session monitoring but impose significant deployment complexity and exhibit false positive rates that may be unacceptable for consumer applications. In contrast, network and request-level behavioral profiling achieves 1.3% false positive rate, sub-5-second mean detection latency, and 4.2 ms per-request overhead with no machine learning requirements and straightforward server-side deployment.

The hybrid framework that combines phishing-resistant WebAuthn initial authentication with lightweight continuous session validation addresses both temporal phases of the authentication attack surface within a unified security architecture. This two-layer approach satisfies all five security objectives derived from the formal adversary model. Open research challenges identified by this review include threshold personalization for heterogeneous user populations, fingerprinting evasion resilience, mobile platform integration, machine learning enhancement, standardization of session monitoring interfaces, and privacy-preserving monitoring designs. Progress on these challenges will be essential to realizing comprehensive, end-to-end authentication security for next-generation web applications.

ACKNOWLEDGEMENT

The authors acknowledge the Department of Computer Science & Engineering, C.G.P.I.T., Uka Tarsadia University, Bardoli, Gujarat, India for institutional support during this research.

REFERENCES

- [1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in Proc. IEEE Symp. Security Privacy, May 2012, pp. 553-567.
- [2] A. Shostack, *Threat Modeling: Designing for Security*. New York, NY, USA: Wiley, 2014.
- [3] D. Fett, R. Kusters, and G. Schmitz, "A comprehensive formal security analysis of OAuth 2.0," in Proc. ACM Conf. Comput. Commun. Security (CCS), Oct. 2016, pp. 1204-1215.
- [4] FIDO Alliance and W3C, "Web Authentication: An API for accessing public key credentials," W3C Recommendation, Apr. 2021. [Online]. Available: <https://www.w3.org/TR/webauthn-2/>
- [5] M. Alaca and P. C. van Oorschot, "Device fingerprinting for augmenting web authentication: Classification and analysis of methods," in Proc. Annual Comput. Security Appl. Conf. (ACSAC), Dec. 2016, pp. 289-301.
- [6] A. Parmar and R. Patel, "HPRAF-CA: Hybrid phishing-resistant authentication framework with continuous session validation: Design, threat modeling, and empirical analysis," Dept. of CSE, C.G.P.I.T., Uka Tarsadia University, Bardoli, Gujarat, India, 2025.
- [7] D. Hardt, "The OAuth 2.0 authorization framework," Internet Engineering Task Force, RFC 6749, Oct. 2012. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc6749>
- [8] M. Jones, J. Bradley, and N. Sakimura, "JSON web token (JWT)," Internet Engineering Task Force, RFC 7519, May 2015. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7519>
- [9] FIDO Alliance, "Client to Authenticator Protocol (CTAP)," FIDO Alliance Specification, Jan. 2022. [Online]. Available: <https://fidoalliance.org/specs/fido-v2.1-ps-20210615/>
- [10] NIST, "Digital identity guidelines: Authentication and lifecycle management," NIST SP 800-63B, Jun. 2017. [Online]. Available: <https://pages.nist.gov/800-63-3/sp800-63b.html>
- [11] E. Bursztein et al., "Handcrafted backdoors in deep neural networks," in Proc. IEEE Symp. Security Privacy, May 2014, pp. 468-482.

- [12] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, "OpenID Connect Core 1.0," OpenID Foundation Specification, Dec. 2023. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html
- [13] C. Mainka, V. Mladenov, J. Schwenk, and T. Wich, "SoK: Single sign-on security — An evaluation of OpenID Connect," in Proc. IEEE Eur. Symp. Security Privacy, Apr. 2017, pp. 231-246.
- [14] S. Gajek, T. Leinweber, H. Schwenk, and J. Schwenk, "On the security of modern single sign-on protocols: Second-order vulnerabilities in OpenID Connect," IEEE Trans. Inf. Forensics Security, vol. 14, no. 8, pp. 2157-2171, Aug. 2019.
- [15] S. Sivakorn, I. Polakis, and A. D. Keromytis, "The cracked cookie jar: HTTP cookie hijacking and the exposure of private information," in Proc. IEEE Symp. Security Privacy, May 2016, pp. 251-266.
- [16] OWASP Foundation, "OWASP API Security Top 10: 2023," OWASP Project, 2023. [Online]. Available: <https://owasp.org/API-Security/editions/2023/en/Ox00-header/>
- [17] A. Shostack, "Experiences threat modeling at Microsoft," in Proc. MODSEC@MoDELS, 2008.
- [18] A. Laszka, M. Felegyhazi, and L. Buttyan, "A survey of interdependent information security games," ACM Comput. Surv., vol. 47, no. 2, pp. 1-38, Jan. 2015.
- [19] NIST, "Zero trust architecture," NIST SP 800-207, Aug. 2020. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf>
- [20] S. Park and A. U. Usman, "Survey of attack surface and countermeasures in zero trust architecture," J. Netw. Comput. Appl., vol. 205, May 2022, Art. no. 103446.
- [21] C. Mainka, V. Mladenov, and J. Schwenk, "Do not trust me: Using malicious IdPs for analyzing and attacking Single Sign-On," in Proc. IEEE Eur. Symp. Security Privacy, 2016, pp. 321-336.
- [22] S. Goel and R. A. Shawky, "Estimating the market impact of security breach announcements on firm values," Inf. Manage., vol. 46, no. 7, pp. 404-410, Oct. 2009.
- [23] OWASP Foundation, "OWASP Top Ten Web Application Security Risks," OWASP Project, 2021. [Online]. Available: <https://owasp.org/www-project-top-ten/>
- [24] P. A. Grassi, M. E. Garcia, and J. L. Fenton, "Digital identity guidelines," NIST SP 800-63-3, Jun. 2017.
- [25] D. Basin et al., "A formal analysis of 5G authentication," in Proc. ACM Conf. Comput. Commun. Security (CCS), Oct. 2018, pp. 1383-1396.