International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Federated Identity Security: Challenges in SAML and OIDC Implementations

Sharath Chandra Thurupati

Abstract: Federated identity management enables seamless, credential-free authentication across organizational boundaries, yet its practical implementation introduces a complex and often underappreciated attack surface. This paper presents a structured security analysis of the two dominant federation protocols — Security Assertion Markup Language (SAML) 2.0 and OpenID Connect (OIDC) — examining their architectural vulnerabilities, real-world misconfiguration patterns, and the operational challenges encountered in enterprise deployments. Drawing on direct implementation experience with IBM Security Verify Access and WebSEAL across large-scale financial and telecommunications environments, the paper analyzes four representative failure scenarios: SAML assertion signature validation failure due to certificate mismatch, clock skew-induced timestamp invalidation, redirect loop misconfiguration, and OIDC JSON Web Key Set endpoint validation failure [9]. For each scenario, root cause analysis, detection methodology, and corrective configuration are presented in reproducible detail. A vulnerability taxonomy covering assertion manipulation, token replay, trust relationship failures, and misconfiguration risks is developed and mapped to protocol-specific mitigations. Comparative security evaluation of SAML and OIDC across five dimensions — assertion integrity, token security, configuration attack surface, debugging complexity, and Zero Trust alignment — demonstrates that neither protocol is universally superior; rather, protocol selection and hardening strategy must be driven by the specific deployment context. The paper concludes with a set of actionable best practices for secure federation design, certificate lifecycle management, and continuous monitoring in enterprise Identity and Access Management environments.

Keywords: *Federated Identity, SAML 2.0, OpenID Connect, JSON Web Token, WebSEAL, Identity Federation, Single Sign-On Security, Cybersecurity*

1. Introduction

Federated identity management has become the foundational authentication architecture for modern enterprise environments. By enabling users to authenticate once with a trusted Identity Provider (IdP) and access multiple applications across organizational boundaries without re-submitting credentials, federation reduces the attack surface associated with password proliferation, centralizes identity governance, and substantially improves the user experience in multi-application, multi-organization environments [1]. The two protocols that dominate enterprise federation deployments are Security Assertion Markup Language (SAML) 2.0, an XML-based standard widely used in enterprise Single Sign-On (SSO) scenarios, and OpenID Connect (OIDC), a modern identity layer built on top of the OAuth 2.0 authorization framework that has become the standard for cloud, mobile, and API-centric applications. Both protocols have achieved broad adoption and carry decades of implementation experience; both also introduce distinctive security challenges that, when inadequately addressed, can render federation deployments more dangerous than the fragmented authentication architectures they replaced.

The security challenges inherent in federation are not primarily theoretical: they are operational. The most consequential vulnerabilities encountered in enterprise federation environments arise not from cryptographic weaknesses in the underlying protocols but from implementation errors, configuration mistakes, trust relationship failures, and the operational complexity of maintaining synchronized, certificate-valid federation relationships across systems owned and administered by different teams. A misconfigured Service Provider (SP) metadata endpoint, an expired signing certificate, a clock skew of a few minutes between federated systems, or an incorrectly specified callback URL can each cause authentication failures or, more critically, open security gaps exploitable by adversaries with access

MSR Technology Group, USA

to network traffic or the ability to manipulate assertion content [2]. These failure modes are not edge cases: they represent the dominant categories of federation-related incidents encountered in production enterprise environments.

This paper addresses this operational reality directly. The primary contribution is a structured analysis of federation security failures grounded in implementation experience with IBM Security Verify Access (ISVA) and WebSEAL in enterprise environments supporting tens of thousands of federated users. The paper identifies the principal vulnerability classes in both SAML 2.0 and OIDC implementations, analyzes four specific real-world failure scenarios with root cause identification and corrective configuration, develops a comparative security evaluation across five dimensions, and derives a set of actionable best practices for secure federation design. This paper makes the following contributions: (1) a protocol-specific vulnerability taxonomy covering the primary attack and failure categories for SAML 2.0 and OIDC; (2) detailed root-cause analysis and resolution for four representative enterprise deployment failures; (3) a comparative five-dimension security assessment of SAML and OIDC; and (4) a practitioner-validated set of security controls and monitoring practices for enterprise federation environments. The paper is structured as follows: Section 2 reviews the relevant literature; Section 3 provides an architectural overview of the two protocols; Section 4 analyzes the principal security vulnerabilities; Section 5 presents the enterprise deployment scenarios; Section 6 describes best practices; Section 7 presents comparative evaluation results; Section 8 discusses challenges; and Section 9 concludes.

The research gap addressed by this paper is the absence of a practitioner-grounded, deployment-experience-based security analysis of SAML and OIDC that integrates protocol-level vulnerability characterization with enterprise operational scenarios. The existing literature addresses protocol security at a theoretical or specification level, or focuses on specific attack classes in isolation. The integrated, operational perspective presented here — covering vulnerability taxonomy, failure scenario analysis, comparative evaluation, and best practice derivation in a single coherent framework — represents a contribution to the IAM security literature not currently available in existing publications.

2. Related Work

The security properties of SAML and OAuth-family protocols have been studied extensively in the research literature, though the focus has predominantly been on formal protocol analysis and specific attack classes rather than on the integrated operational challenges that characterize enterprise deployments. Naik and Jenkins [1] provided an early comparative assessment of SAML, OAuth 2.0, and OIDC from a cloud federated identity management perspective, evaluating each against architectural design, security strength, and security vulnerability dimensions. Their work established a foundational taxonomy of federation protocol trade-offs that remains relevant to deployment decisions. More recent work has examined specific protocol vulnerabilities: Li and Mitchell [3] analyzed the privacy implications of user access tracking by Identity Providers in OAuth 2.0 and OpenID Connect deployments, demonstrating that IdPs can construct detailed access profiles of user behavior across all federated Service Providers — a finding with significant implications for enterprise privacy governance.

The security of JSON Web Token (JWT)-based authentication, central to OIDC, has received attention both in the academic literature and in practical security research. The OpenID Foundation disclosed a vulnerability (CVE-2025-27370) in the private_key_jwt client authentication specification in 2025, revealing that specification ambiguities in JWT audience handling could be exploited under certain configuration conditions [4]. This disclosure underscores the importance of following current errata-incorporating specifications rather than legacy implementations. Complementary work on SAML assertion security has examined XML signature wrapping attacks, assertion replay, and the consequences of improper schema validation — attack classes that remain practically relevant in enterprise deployments where SAML implementations have not been updated to reflect current hardening guidance [2].

The role of the WebSEAL reverse proxy in enterprise federation architectures has been examined primarily in deployment-oriented technical literature rather than in peer-reviewed academic publications, reflecting the proprietary nature of the platform. However, the general class of security enforcement mechanisms implemented by WebSEAL — policy enforcement at the network

ingress point, junction-based access control, and integration with directory-backed identity stores — has been studied in the context of enterprise access management architectures [5]. The framework presented in this paper extends this body of work by providing a detailed operational analysis of federation failure modes within a WebSEAL-based deployment architecture, contributing scenario-level technical detail not available in existing published work.

Machine learning-based approaches to detecting federation anomalies and access management threats have emerged as a complementary layer to protocol-level security controls. AI-driven anomaly detection for IAM in cloud platforms, employing Long Short-Term Memory Autoencoder (LSTM-AE) models, has demonstrated an accuracy of 0.997 and an F1 score of 0.998 on IAM event logs, establishing the viability of behavioral anomaly detection as a real-time monitoring layer for federated authentication systems [6]. The integration of such detection capabilities with SAML and OIDC event logging pipelines represents a productive direction for enterprise security operations, building on the monitoring and logging best practices described in Section 6 of this paper.

3. Overview of Federation Protocol Architectures

3.1 SAML 2.0 Architecture and Assertion Flow

Security Assertion Markup Language 2.0 is an XML-based open standard for exchanging authentication and authorization data between an IdP and a Service Provider [10]. The fundamental exchange unit in SAML is the assertion — an XML document signed by the IdP using a private key certificate, containing authentication statements that attest to the identity of the authenticated user, optionally accompanied by attribute statements and authorization decision statements [15]. The SAML Web Browser SSO profile defines the canonical enterprise SSO flow: the user requests access to a resource at the SP; the SP generates an authentication request and redirects the browser to the IdP; the IdP authenticates the user and generates a signed SAML assertion; the assertion is transmitted back to the SP via a browser-mediated POST binding; the SP validates the assertion signature against the IdP's public key certificate, checks the assertion's validity window, verifies the recipient and audience fields, and if all checks pass, establishes a session for the user. The security of this exchange depends entirely on the integrity of the assertion signature, the validity of the certificate used to verify it, and the correctness of the SP's validation logic.

The key security properties of SAML — strong cryptographic assertion signing, a mature and widely implemented standard, and a rich attribute statement model — are accompanied by well-characterized limitations: the XML structure is inherently complex and error-prone to implement, XML signature handling is technically demanding, and debugging authentication failures requires detailed assertion-level inspection that many toolchains do not provide transparently. The debugging and maintenance overhead of SAML-based federation scales with the number of federated partners and the complexity of the attribute mapping requirements, creating operational complexity that in practice leads to the misconfigurations analyzed in Section 5 [2].

3.2 OpenID Connect Architecture and Token Flow

OpenID Connect is a lightweight identity layer built on top of the OAuth 2.0 authorization framework, using JSON Web Tokens (JWTs) as the primary assertion format [8]. The OIDC authorization code flow — the recommended flow for server-side web applications — operates as follows: the user authenticates at the IdP (acting as the OIDC Provider, OP); the OP issues an authorization code to the Relying Party (RP) via browser redirect; the RP exchanges the code for an ID token and optionally an access token at the OP's token endpoint; the RP validates the ID token's signature using the OP's public key obtained from the JSON Web Key Set (JWKS) endpoint, verifies the issuer, audience, and expiration claims, and if all checks pass, establishes an authenticated session. The compactness of JWT compared to XML SAML assertions, combined with the REST-based token exchange, makes OIDC substantially easier to implement and debug, and better suited to mobile and API-centric architectures [3].

The security of OIDC depends on the correct validation of ID token claims — particularly issuer (iss), audience (aud), expiration (exp), and nonce — and on the secure handling of tokens in transit and at rest. Token replay attacks, audience confusion

vulnerabilities, and JWKS endpoint misconfiguration are the primary security risks in OIDC deployments. The audience confusion class of attacks, in which a token issued for one RP is accepted by another RP that does not validate the audience claim correctly, is a consequence of improper implementation that has been observed in production deployments and documented in formal analysis [4]. The comparison between SAML and OIDC security properties across the five evaluation dimensions is presented in Section 7.

4. Security Vulnerabilities in Federation Protocols

4.1 Assertion Manipulation in SAML

SAML assertion manipulation encompasses a class of attacks in which an adversary modifies the content of a SAML assertion after it has been issued by the IdP, with the goal of altering the authentication outcome at the SP. The most well-known variant is the XML signature wrapping (XSW) attack, in which the attacker exploits ambiguities in the XML Document Object Model (DOM) traversal logic used by the SP's assertion parser to present a modified assertion body while preserving a valid signature over the original content. The attack is viable when the SP validates the signature over one node of the XML document but applies the assertion content from a different node — a condition that can arise in implementations that use naive XPath queries or non-standard XML parsing libraries. Prevention requires strict XML schema validation before signature verification, use of exclusive canonicalization as specified in the SAML 2.0 standard, and enforcement of assertion subject and attribute constraints [10].

Assertion replay attacks represent a second manipulation class: an adversary captures a valid SAML assertion and resubmits it to the SP before the assertion's validity window expires, gaining unauthorized authenticated access using a credential that was legitimately issued to a different user or in a different session context. Prevention requires SP-side assertion ID tracking — maintaining a short-term cache of processed assertion IDs and rejecting assertions with previously seen IDs — combined with tight validity window enforcement. In WebSEAL-based deployments, the junction configuration and Advanced Access Control policies must be explicitly configured to enforce assertion ID uniqueness; this is not applied by default and must be verified as part of the federation security review.

4.2 Token Vulnerabilities in OIDC

OIDC token security centers on the integrity and controlled lifespan of the JWT ID token. The most operationally prevalent vulnerability class is improper token validation: an RP that fails to verify the token signature, does not check the issuer claim, or accepts tokens with expired expiration timestamps is exposed to forgery and replay attacks that can result in unauthorized session establishment. The OpenID Connect Core 1.0 specification [7] mandates a specific set of ID token validation steps — signature verification, issuer check, audience check, expiration check, and nonce verification, where applicable — and implementations that omit or incorrectly implement any of these steps create exploitable conditions. Verification of JWKS endpoint configuration is particularly critical: an RP that caches a stale public key set may fail to validate tokens issued after a key rotation, while an RP that accepts tokens signed with a "none" algorithm exploits the algorithm confusion vulnerability class documented in the JWT security literature [13].

Token replay in OIDC is mitigated through the use of short-lived tokens (typically 300–600 seconds for ID tokens), nonce binding in the authorization request, and the use of the Proof Key for Code Exchange (PKCE) extension for public clients. In enterprise OIDC deployments using WebSEAL as the RP, token validation is implemented through the federation configuration on the WebSEAL Policy Server, and the JWKS endpoint must be explicitly configured and periodically refreshed. Production deployments should enforce automatic JWKS key rotation with a rotation interval of no more than 90 days, and RP configurations should be validated against the current JWKS endpoint after each rotation to prevent the validation failures analyzed in Section 5.

4.3 Misconfiguration Risks

Misconfiguration is empirically the most prevalent root cause of federation security failures in enterprise environments. The attack surface created by misconfiguration spans both protocols: incorrectly specified assertion consumer service endpoints in SAML SP metadata allow assertion

delivery to unintended destinations; missing or incorrect certificate entries in IdP metadata cause signature validation failures that may be resolved by temporarily disabling signature checking — a dangerous remediation that eliminates the protocol's primary integrity guarantee; improperly configured OIDC redirect URIs allow authorization code interception via open redirect vulnerabilities; and incorrect clock synchronization between federated systems causes time-based assertion and token validity checks to fail intermittently, leading to the erratic authentication behavior described in the deployment scenarios in Section 5. The operational complexity of managing federation configurations across multiple organizational boundaries, each with independent change management processes, creates persistent misconfiguration risk that technical controls alone cannot fully mitigate [1].

4.4 Trust Relationship Failures

Federation trust is binary and certificate-dependent: either the SP trusts the IdP's signing certificate and can validate assertions, or it does not, and authentication fails. This binary nature creates fragility at the trust boundary: certificate expiration, key rotation without metadata update, and clock desynchronization between systems can each cause complete authentication outages without any underlying security event. In multi-organization federation environments — where the IdP and SP are operated by different entities with independent certificate lifecycle processes — these failures are not uncommon. For enterprise federation incidents, certificate issues related to trust are among the top two causes of unplanned production outages associated with federation. The certificate mismatch issue is second only to network connectivity issues between federation partners [1]. The four enterprise federation deployment scenarios presented in Section 5 include two additional scenarios that are derived from trust relationship issues (the certificate mismatch issue and the clock skew issue) as these are common in enterprise federation scenarios.

Table 1 — Vulnerability taxonomy for SAML 2.0 and OIDC federation protocols with protocol-specific mitigations [2][3][4][7][10][13]

Vulnerability class	Attack variant	Protocol	Severity	Primary mitigation
Assertion manipulation	XML signature wrapping (XSW)	SAML	High	Strict XML schema validation before signature check; exclusive canonicalization; subject and attribute constraint enforcement
	Assertion replay	SAML	High	SP-side assertion ID cache; validity window ±5 min; WebSEAL junction configured to enforce assertion ID uniqueness
Token vulnerabilities	Improper token validation	OIDC	High	Mandatory validation of iss, aud, exp, nonce, and signature per OpenID Connect Core 1.0
	Algorithm confusion	OIDC	High	Explicitly disallow "none" algorithm; constrain accepted algorithms to RS256 or ES256
	Token replay	OIDC	Medium	Short-lived tokens (300–600 s); nonce binding; PKCE for all public clients
Trust relationship failures	Certificate mismatch	SAML	High	Automated metadata refresh (24 h); certificate expiry alerts at 60- and 30-day thresholds

	JWKS key rotation outage	OIDC	High	Automatic JWKS refresh with force-refresh on failure; key overlap window during rotation
Misconfiguration risks	Incorrect ACS URL / redirect loop	Both	High	Pre-deployment validation checklist; junction endpoint exclusion from authentication interception
	Clock skew	SAML	Medium	NTP synchronization enforced across all federation systems; validity window ±5 min post-alignment
	Audience confusion	OIDC	High	Strict aud claim validation at every RP; reject tokens where aud ≠ registered client_id

5. Enterprise Deployment Scenarios: WebSEAL Implementation Analysis

5.1 Scenario 1 — SAML Assertion Signature Validation Failure

Observation: In a financial services federation environment, users could not authenticate using SAML SSO after rotating the IdP certificate on schedule. The IdP issued SAML assertions with no authentication errors, but the WebSEAL instance reported "Signature validation failed" errors for all incoming SAML assertions in the WebSEAL logs. Analysis: The IdP had rotated its signing certificate as part of a regular key management process, replacing it with a new signing certificate and new signing material for signing outgoing assertions. The SP metadata held by WebSEAL, however, still referenced the old signing certificate from the IdP. WebSEAL's assertion validator rejected all the assertions because the new signature's certificate was not present in the SP's cached metadata. The IdP's certificate rotation had not been communicated to the SP federation team, and there was no way to refresh the SP's metadata automatically. Resolution: The SP metadata was manually updated with the new signing certificate from the IdP. As a follow-up, an automated job was scheduled to refresh the SP metadata from the IdP's published metadata URL every 24 hours. An alerting mechanism was configured to notify the federation operations team when certificate validity windows fall below 60 days. Lesson: Certificate lifecycle management must be treated as a shared operational responsibility across all federation partners, with automated metadata refresh as the primary mitigation for uncoordinated rotation events.

This scenario illustrates a structural vulnerability in enterprise federation: the dependency on correct, current SP metadata is absolute, yet metadata updates are frequently manual and reactive rather than automated and proactive. In IBM Security Verify Access deployments, the WebSEAL federation configuration supports metadata refresh via scheduled policy reloads, but this capability must be explicitly configured; the default posture is static metadata. Organizations should implement automated metadata refresh for all federation partnerships as a baseline operational control, and should test metadata refresh procedures as part of regular federation DR exercises.

5.2 Scenario 2 — Clock Skew and Timestamp Validation Failure

Observation: Users in a telecommunications enterprise environment experienced intermittent SAML authentication failures affecting approximately 15% of login attempts, with no pattern in the affected user population. WebSEAL logs show error messages "Assertion not yet valid" and "Assertion has expired" for these requests. Background: SAML assertions contain "NotBefore" and "NotOnOrAfter" timestamp attributes that specify the time range when the assertion is valid. This window of time is usually ± 5 minutes from the issue time. Because the enterprise had not deployed NTP time synchronization between all of the systems in the federation, several servers (including several WebSEAL instances) had computed their clocks to be 4 to 7 minutes different from the

canonical time. The IdP sent assertions with valid timestamps; however, when fed into WebSEAL instances with incorrect clocks, they were seen as not yet valid (if the WebSEAL clock was ahead) or as expired (if the WebSEAL clock was behind), leading to the rejection of approximately 15% of the assertions. Resolution: All impacted federation environments were synchronized via NTP to an authoritative NTP server. The IdP configuration assertion validity window was temporarily widened from ±5 to ±10 minutes until the NTP was completely synchronized. After verification, the IdP window was returned to ±5 minutes. From this experience, proper time synchronization should be an assumed prerequisite for SAML-based federation and should go on the federation deployment checklist.

5.3 Scenario 3 — Redirect Loop Misconfiguration

Fact: After changing a WebSEAL junction protecting a financial application, users who attempt to perform SAML federation authentication are stuck in a loop between the SP site and IdP site, without successfully authenticating. Observation: A configuration change to the WebSEAL junction inadvertently changed the federation's assertion consumer service (ACS) URL in the SP metadata to an incorrect value. Unfortunately the path WebSEAL was configured to protect was not an assertion delivery endpoint, but a path WebSEAL was configured to recognize and redirect to the IdP. The IdP sent an assertion to the ACS URL. WebSEAL did not recognize it as being a location where assertions should be delivered, so it asked for a new authentication request from the IdP, which delivered a new assertion, and the cycle repeated. Resolution: Update the SP metadata's ACS URL value to point to the federation endpoint of the WebSEAL junction. Implementation of an alternate path in the federation endpoint to avoid authentication interception for the federation endpoint paths was also considered. A pre-deployment checklist was created to validate the correctness of the ACS URL before making a change in production. Lesson: Redirect loop scenarios are a direct consequence of ACS URL misconfiguration and are completely avoidable with systematic pre-deployment validation of the ACS URL.

5.4 Scenario 4 — OIDC JWKS Endpoint Validation Failure

Observation: An OIDC-enabled application behind WebSEAL stopped accepting ID tokens after a planned OP key rotation. WebSEAL logs reported "Token signature validation failed" errors. Analysis: The OP completed a key rotation and updated the JWKS endpoint with the new signing key pair. However, the JWKS endpoint had not yet been consumed by WebSEAL to get the new public key. WebSEAL's OIDC federation configuration had cached the previous well-known jwks_uri metadata content, but WebSEAL was not configured to automatically refresh the key set from the jwks_uri. As a result, WebSEAL attempted to validate ID tokens (which were signed with the newly registered key) against the old cached public key, and all such validations failed. Resolution: The OIDC properties on the WebSEAL Policy Server were configured with the refresh interval set to 24 hours, and an option for refresh when signature validation fails. The WebSEAL outage was resolved by clearing the JWKS cache in WebSEAL, and forcing a refresh from the OP's JWKS endpoint. Post-incident, a monitoring check was added to alert if the JWKS endpoint became unavailable or if the keys need to be refreshed every hour. This incident has also highlighted the need to configure automatic refresh of the JWKS for OIDC deployments, and end-to-end testing of federation configurations whenever the OP rotates keys.

6. Best Practices for Secure Federation Design

6.1 Certificate and Key Lifecycle Management

The most important operational security control in a SAML-based federation is certificate management: certificates used to sign SAML assertions must be rotated (usually annually), and in a coordinated fashion with all federation partners to allow all parties sufficient time to update the SP metadata files prior to expiration of certificates. This mechanism should be specified in the federation partnership agreements. Ideally, refresh should be automated. In IBM Security Verify Access deployments, the federation configuration on the Policy Server should use the published metadata URL of the IdP rather than a static metadata file, so that certificates and endpoints are updated automatically with minimal manual intervention. Certificate validation checks should be implemented, which will flag certificates that are expiring within 60 or 30 days (or less) for all certificates in federation trust chain. OIDC key

rotation must be implemented with overlap so that old and new keys are both present in the JWKS endpoint at the same time when the key is rotated to avoid a key not being found.

6.2 Configuration Validation and Change Control

Federation configurations must be subject to rigorous pre-deployment validation to prevent the misconfiguration failures documented in Section 5. A federation configuration validation checklist should cover, at minimum: ACS URL accuracy and WebSEAL junction endpoint alignment; SP metadata completeness, including entity ID, signing certificate, and encryption certificate where applicable; OIDC redirect URI registration at the OP; JWKS endpoint accessibility and key set currency; NTP synchronization status across all federation-participating systems; and end-to-end authentication flow testing in a staging environment before production deployment. Any federation configuration change — however minor — should require completion of this checklist, as the scenarios in Section 5 demonstrate that consequential failures frequently arise from changes that appear low-risk in isolation.

6.3 Token and Assertion Security Controls

Protocol-level security controls must be applied consistently and completely. For SAML: assertion signature verification must never be disabled, even as a temporary troubleshooting measure; XML schema validation must precede signature verification to prevent XSW attacks; assertion ID tracking must be implemented to prevent replay; and validity window enforcement should use tight time bounds (±5 minutes maximum, with NTP synchronization confirmed). For OIDC: ID token validation must implement all mandatory steps defined in OpenID Connect Core 1.0 [7] — signature verification, issuer check, audience check, expiration check, and nonce verification; the "none" algorithm must be explicitly disallowed in token validation configuration; access tokens must use short validity windows (300–600 seconds); and PKCE must be enforced for all public clients. The controls must be included within every federation configuration review and penetration test.

6.4 Monitoring, Logging, and Anomaly Detection

For security monitoring and operational troubleshooting purposes, it is advisable to log federation authentication events by enabling the WebSEAL trace logging feature and setting the trace level to a value that will log the contents of assertions, the outcome of validations, and the creation of sessions. Logs should be sent to a Security Information and Event Management (SIEM) system and correlated in real-time with alerts. Anomaly detection models applied to federation event logs — as demonstrated by LSTM-AE approaches achieving 0.997 accuracy on IAM event data [6] — can identify authentication anomalies not detectable through static rule-based monitoring, including assertion replay patterns, impossible travel indicators, and unusual assertion attribute value distributions. The combination of protocol-level controls and behavioral monitoring provides the defense-in-depth posture appropriate for enterprise federation environments supporting sensitive applications.

Table 2 — Best practices for secure federation design: controls, applicability, and implementation guidance [1][2][5][6][7][9][12]

Category	Control	Applies to	Implementation guidance
Certificate and key lifecycle	Automated metadata / JWKS refresh	Both	24 h refresh interval; force-refresh on OIDC validation failure; key overlap window during rotation

	Certificate expiry monitoring	SAML	Alerts at 60- and 30-day thresholds; rotation coordination codified in partnership agreements
	Scheduled key rotation	Both	SAML certificates annually; OIDC keys at ≤90-day intervals with key overlap
Configuration validation	Pre-deployment checklist	Both	Verify ACS URL, SP metadata, redirect URIs, JWKS accessibility, NTP status, and end-to-end staging test before production
	NTP synchronization	SAML	Enforce common authoritative NTP source across all federation-participating systems
Token and assertion security	Complete ID token validation	OIDC	All steps per OpenID Connect Core 1.0: signature, iss, aud, exp, nonce; disallow "none" algorithm
	PKCE + short token lifetimes	OIDC	PKCE for all public clients; access token validity 300–600 s
	Assertion signature + ID tracking	SAML	Never disable signature verification; assertion ID cache to prevent replay; configured explicitly in WebSEAL AAC
Monitoring and anomaly detection	WebSEAL trace logging to SIEM	Both	Capture assertion content, validation results, session events; forward to SIEM for real-time correlation
	ML-based anomaly detection	Both	LSTM-AE models on federation event logs; targets: replay patterns, impossible travel, unusual attribute distributions (accuracy 0.997 [6])

7. Comparative Security Evaluation: SAML versus OIDC

The comparative evaluation of SAML and OIDC across five security dimensions — assertion integrity, token security, configuration attack surface, debugging complexity, and Zero Trust alignment — is presented in this section. The analysis draws on the protocol specifications, vulnerability analysis in Section 4, and deployment scenarios in Section 5, and provides recommendations for protocol design and hardening when deployed in enterprise federation scenarios.

Assertion integrity: SAML assumes XML digital signatures over the whole assertion document, leading to cryptographically strong assurance provided that exclusive canonicalization, strict schema validation and XML Signature Wrapping attacks are properly reduced. OIDC has a slightly stronger story for assertion integrity signature. Signing a JWT is just easier. OIDC's signing algorithms also have an algorithm confusion attack

if the RP doesn't specify its supported signing algorithms. Both protocols have sufficient properties for assertion integrity, but SAML is more complex and more susceptible to configuration errors. JWT tokens are smaller and more debuggable than SAML. They are also introspectable, but all claims must be validated. SAML is larger and more opaque, but has been used in enterprise environments for decades, and validators have had decades of hardening. OIDC wins on token expiry granularity, while SAML wins on enterprise maturity. Configuration attack surface: Both protocols can suffer from configuration errors. SAML has a much larger attack surface (SP metadata, certificate handling and rotation, assertion consumer service URLs, and attribute mapping), but just like OIDC's redirect URI registries, JWKS endpoints, and configuration of validation rules if misconfigured, it is equally dangerous. Debugging: OIDC allows for inline inspection of the JWT by trivially decoding it, and the HTTP-based token exchange can be inspected with normal network analysis tooling. SAML's XML based assertion model requires parsing, and the content of the assertion is obfuscated in the browser-POST binding. Zero Trust alignment: OIDC's token-based and API optimized model mapped to Zero Trust access control models, particularly in micro-service and cloud-native application environments [12]. SAML, in contrast, is still the most extensively used SSO technology in enterprise environments, and is less favorably disposed to Zero Trust's continuous identity verification model. Nevertheless, a ZTA can be based on SAML with policies for re-evaluation, as in ISVA [5].

Quantitative summary: Based on the five-dimensional assessment, OIDC is rated higher than SAML on debugging complexity (4/5 vs. 2/5) and Zero Trust alignment (4/5 vs. 3/5); SAML is rated higher on assertion integrity maturity (4/5 vs. 3/5). Both protocols are rated 3/5 on configuration attack surface and token security, reflecting comparable misconfiguration risk and the importance of implementation correctness in both cases. Neither protocol is unambiguously superior; the evaluation supports a deployment strategy in which OIDC is preferred for new cloud-native integrations while SAML is retained for legacy enterprise SSO integrations where its maturity and existing toolchain investment provide operational advantage. The comparative data are presented in detail in Table 1 of the companion workbook.

Table 3 — Comparative security evaluation of SAML 2.0 and OpenID Connect across five dimensions [1][2][3][7][12]

Dimension	SAML (/5)	OIDC (/5)	Advantage	Key basis
Assertion integrity	4	3	SAML	XML digital signatures; OIDC algorithm confusion risk
Token security	3	3	Equal	Equivalent risk; both require correct implementation
Configuration attack surface	3	3	Equal	SAML surface broader; OIDC equally consequential

Debugging complexity	2	4	OIDC	JWT decoded inline; SAML XML POST binding opaque
Zero Trust alignment	3	4	OIDC	OIDC token model maps to continuous verification

8. Challenges and Open Problems

Several challenges persist in enterprise federation security that are not fully resolved by the controls described in this paper. Multi-organization trust governance presents a coordination problem that technical controls alone cannot resolve: in federations involving multiple organizations — each with independent security policies, certificate lifecycle processes, and change management procedures — the operational discipline required to maintain synchronized, secure federation configurations is difficult to sustain over time. Federation partnership agreements that specify technical security requirements, mandatory certificate rotation timelines, and incident communication procedures provide a governance framework, but enforcement depends on the commitment of each participating organization to uphold the agreed standards [1].

Debugging complexity, particularly for SAML-based federations, remains an open operational problem. The browser-mediated assertion delivery mechanism obscures assertion content from standard monitoring tools; assertion decoding and validation logging require either specialized tooling or deep familiarity with the ISVA trace log format. The development and adoption of standardized federation debugging tooling — analogous to the JWT decoding tools available for OIDC — would substantially reduce the time-to-resolution for federation incidents and lower the expertise barrier for federation operations teams. Until such tooling is broadly available, organizations should invest in detailed runbooks for the most common federation failure scenarios, incorporating the root-cause analysis and resolution patterns documented in Section 5 of this paper.

Zero Trust integration for SAML-based federations remains an area of active development. The session-based trust model implicit in SAML SSO — where a successful assertion validation establishes a trust relationship persisting for the duration of the session — is fundamentally in tension with the continuous verification mandate of Zero Trust Architecture. Extending SAML-based federation to support continuous re-evaluation of the trust assertion, triggered by behavioral anomaly signals from the monitoring layer, requires architectural extensions to the standard federation profile that are not yet widely implemented [11]. ISVA's Advanced Access Control module provides a partial path to this capability through context-aware re-authentication policies, but a complete Zero Trust-aligned federation architecture for SAML environments remains an open research and engineering problem.

9. Conclusion

In summary, this paper described our security analysis of SAML 2.0 and OIDC federation implementations in enterprise Identity and Access Management based on our practical experiences on deploying and operating IBM Security Verify Access and WebSEAL federation components. It presented four representative federation failure scenarios involving SAML assertion signature verification failure (certificate mismatch), timestamp expiration due to clock skew, redirect loop due to federation misconfiguration, and OIDC JWKS endpoint verification failure. Root causes, resolution and operational lessons learned are discussed which comprise the most commonly reported federation security failures encountered in production. A vulnerability taxonomy covering assertion manipulation, token replay, trust relationship failures, and misconfiguration risks was developed and mapped to protocol-specific mitigations.

The comparative evaluation of SAML and OIDC across five security dimensions establishes that protocol selection must be driven by deployment context rather than a single universal security ranking: OIDC is better suited to cloud-native, API-centric, and Zero Trust-aligned environments, while SAML retains operational advantages for legacy enterprise SSO integrations where its maturity and established toolchain support provide practical value. Regardless of protocol choice, the security controls that most significantly reduce enterprise federation risk — certificate lifecycle automation, configuration validation, assertion integrity enforcement, and behavioral anomaly monitoring — are operational in nature and require sustained discipline to maintain across the lifetime of federation partnerships.

The best practices derived from this analysis — automated metadata refresh, NTP synchronization enforcement, pre-deployment configuration validation, complete token validation implementation, and SIEM-integrated anomaly detection — are directly applicable to any enterprise IAM environment operating federated identity services. As federated identity continues to extend across cloud, mobile, and API domains, the operational security discipline documented in this paper becomes more, not less, critical. The structured failure scenario analysis and comparative protocol evaluation presented here contribute a practitioner-grounded perspective to the IAM security literature and provide a reference framework for enterprise federation security reviews.

References

[1] N. Naik and P. Jenkins, "Securing digital identities in the cloud by selecting an appropriate federated identity management from SAML, OAuth, and OpenID Connect," in Proc. 2017 11th International Conference on Research Challenges in Information Science (RCIS), Brighton, UK, 2017, pp. 163–174. [Online]. Available: https://ieeexplore.ieee.org/document/7956534/

[2] National Institute of Standards and Technology, "Security and Privacy Controls for Information Systems and Organizations," NIST Special Publication 800-53 Rev. 5, Sep. 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-53r5

[3] W. Li and C. J. Mitchell, "User access privacy in OAuth 2.0 and OpenID Connect," 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Italy, 2020, pp. 664–672. [Online]. Available: https://ieeexplore.ieee.org/document/9229747/

[4] OpenID Foundation, "Notice of a Security Vulnerability," 2025. [Online]. Available: https://openid.net/notice-of-a-security-vulnerability/

[5] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST Special Publication 800-207, Aug. 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-207

[6] B. Rajak et al., "AI-Driven Anomaly Detection for Secure Identity and Access Management in Cloud Platform," 2025 Global Conference in Emerging Technology (GINOTECH), 2024–2025. [Online]. Available: https://ieeexplore.ieee.org/document/11076807/

[7] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, "OpenID Connect Core 1.0 incorporating errata set 2," OpenID Foundation, Dec. 2023. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html

[8] Auth0, "OAuth 2.0 Authorization Framework". [Online]. Available: https://auth0.com/docs/authenticate/protocols/oauth

[9] IBM, "IBM Security Verify Access," 2024. [Online]. Available: https://www.ibm.com/support/pages/system/files/inline-files/verifyaccess_admin_federation_2.pdf

[10] OASIS, "Security Assertion Markup Language (SAML) V2.0 Technical Overview," Mar. 2008. [Online]. Available: https://docs.oasis-

open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

[11] S. Wiefling, J. Tolsdorf, and L. Lo Iacono, "Privacy Considerations for Risk-Based Authentication Systems," 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Vienna, Austria, 2021, pp. 320–327. [Online]. Available: https://ieeexplore.ieee.org/document/9583699/

[12] Cybersecurity and Infrastructure Security Agency Cybersecurity Division, "Zero Trust Maturity Model," CISA, Apr. 2023. [Online]. Available: https://www.cisa.gov/sites/default/files/2023-04/CISA_Zero_Trust_Maturity_Model_Version_2_508c.pdf

[13] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," Internet Engineering Task Force (IETF), May 2015. [Online]. Available: https://doi.org/10.17487/RFC7519

[14] L. Atorf, C. Schorn, J. Rossmann, and C. Schlette, "A framework for simulation-based optimization demonstrated on reconfigurable robot workcells," 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 2017. [Online]. Available: https://doi.org/10.1109/SysEng.2017.8088278

[15] B. Campbell, C. Mortimore, and M. Jones, "Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants," Internet Engineering Task Force RFC 7522, May 2015. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7522