Zentra

Security Whitepaper

Enterprise Identity Platform - Evidence-Based Security Architecture

Version 1.1 | February 2026

FBT (Future Beyond Tech)

Table of Contents

Right-click the TOC and select "Update Field" to refresh page numbers

1. Executive Summary

Zentra is a self-hosted, enterprise-grade OAuth 2.0 and OpenID Connect identity platform designed to secure distributed systems, SaaS applications, and API ecosystems. This whitepaper provides evidence-based security documentation for enterprise security teams, auditors, and procurement professionals.

Document Purpose

This v1.1 whitepaper represents a shift from marketing claims to evidence-based security documentation. All protocol compliance claims are backed by HTTP request/response examples and regression test references. Maturity assessments are calibrated against actual implementation evidence rather than aspirational targets.

Key Improvements in v1.1

Aspect	v1.0	v1.1
Maturity Claim	L4 - Managed (marketing)	L2.5 - Managed (evidence-based)
RFC Compliance	Claims only	HTTP examples + test references
Threat Model	Not included	Complete STRIDE matrix
Residual Risk	Not addressed	Documented with justification
Monitoring	Basic mention	Detailed alert configuration
Governance	Basic	Quarterly reviews, frequencies

Core Positioning

Zentra enables organizations to centralize identity securely while maintaining architectural control and compliance readiness. The platform is built on security-first principles:

- Zero Trust Architecture - No implicit trust between services or users
- Token-based Authentication - Short-lived, cryptographically signed access tokens
- Vendor Independence - Open standards compliance (OAuth 2.0, OIDC, JWT)
- Evidence-Based Claims - All security assertions backed by testable proof
- Defense in Depth - Multiple security layers protecting identity infrastructure

2. Security Philosophy

2.1 Security by Design

Zentra is architected according to established security principles:

Zero Trust Architecture

Every access request is authenticated and authorized, regardless of source. Internal service communication requires the same validation as external requests.

Principle of Least Privilege

Users and services receive only minimum necessary permissions. Scope-based authorization limits blast radius of security incidents.

Defense in Depth

Security controls at network, application, data, and operational layers. No single point of failure.

Secure Defaults

PKCE enforced for public clients, strong cryptography preferred, verbose errors suppressed in production.

2.2 FBT Engineering Security Standards

- No secrets in repository - All credentials externalized
- Structured logging - Parseable formats with security classification
- Clean Architecture - Security isolated from business logic
- CI security scanning - Automated vulnerability detection
- OWASP mitigation - Proactive countermeasures

3. Identity & Authentication Architecture

3.1 Protocol Standards

Protocol	RFC/Standard	Purpose
OAuth 2.0	RFC 6749	Authorization framework
OpenID Connect	OIDC Core	Authentication layer
PKCE	RFC 7636	Code interception prevention
Token Introspection	RFC 7662	Token validation
Token Revocation	RFC 7009	Token invalidation
JWT	RFC 7519	Claims representation
JWK	RFC 7517	Key representation

3.2 Supported Flows

Authorization Code Flow

Primary flow for server-side applications. Authorization code exchanged for tokens via back-channel.

Authorization Code + PKCE

Required for public clients. Prevents authorization code interception via code_verifier validation.

Client Credentials Flow

Machine-to-machine authentication without user involvement.

Refresh Token Flow

Long-lived sessions via token rotation. Each refresh invalidates previous token.

4. Token Security Model

4.1 Access Tokens

- RSA/ECDSA signing - RS256 or ES256 algorithms
- Short lifetime - Recommended <15 minutes
- Audience restriction - Prevents cross-service replay
- Scope embedding - Authorized scopes cryptographically bound

4.2 Refresh Tokens

- Server-side storage - SecurityTokens table with metadata
- Token rotation - New token issued on every refresh
- Revocation on logout - Immediate invalidation
- Device isolation - Per-device token management

4.3 Token Validation

API Gateway Validation

Signature verification using JWKS endpoint. Invalid signatures rejected before backend.

Resource Server Validation

Audience, issuer, and expiry validation. Ensures intended recipient usage.

Introspection Endpoint

Real-time token status checking for opaque tokens or additional validation.

5. Cryptographic Standards

5.1 Key Management

- Asymmetric signing - RSA/ECDSA key pairs
- JWKS endpoint - Public key publication
- Automated rotation - 90-day signing key rotation
- Environment separation - Distinct keys per environment

5.2 Algorithm Support

Algorithm	Type	Status
RS256	RSA + SHA-256	Default
ES256	ECDSA + SHA-256	Optional
HS256	HMAC + SHA-256	Internal only

5.3 Transport Security

- HTTPS enforcement - TLS 1.2+ required
- Secure cookies - Secure and HttpOnly flags
- HSTS headers - Browser protection

6. Authorization & Access Control

6.1 Role-Based Access Control (RBAC)

- Centralized role storage - Consistent enforcement
- Role claims in JWT - Signed role assertions
- API-level enforcement - Resource server validation

6.2 Scope-Based Authorization

- Granular API scopes - Per-endpoint requirements
- Per-client allowed scopes - Explicit authorization
- Gateway enforcement - Scope validation at entry

6.3 Client Restrictions

- Redirect URI validation - Code binding
- PKCE enforcement - Public client requirement
- Grant type restrictions - Flow limitations

7. Multi-Factor Authentication

7.1 Supported Factors

Factor	Type	Implementation
Email OTP	Possession	Time-limited email codes
SMS OTP	Possession	Time-limited SMS codes
Authenticator App	Possession	TOTP (RFC 6238)
LDAP + 2FA	Knowledge + Possession	Enterprise integration

7.2 Security Measures

- Retry limits - Progressive delays
- Account lockout - Temporary suspension
- Configurable policy - Per-group requirements
- Backup codes - Recovery mechanism

8. Audit & Monitoring

8.1 Audit Logging

Comprehensive audit trails for security-relevant events:

- Login events - Success and failure
- Token lifecycle - Issuance, refresh, revocation
- Administrative actions - Role changes, client creation
- MFA events - Enrollment and verification

8.2 Audit Record Fields

Field	Description
Actor	User or service principal
Timestamp	UTC with millisecond precision
Event Type	Security classification
IP Address	Source (if configured)
Correlation ID	Request tracing

8.3 Observability

- Structured logging - JSON format, SIEM compatible
- Correlation IDs - Distributed tracing
- Health endpoints - Liveness/readiness probes
- OpenTelemetry - Distributed tracing (planned)

9. RFC Compliance Evidence

This section provides evidence-based documentation of RFC compliance through actual HTTP request/response examples and regression test references.

9.1 RFC 6749 - OAuth 2.0 Authorization Framework

Authorization Endpoint

Positive Case - Successful Authorization Request:

```
GET
/connect/authorize?response_type=code&client_id=web-app&redirect_uri=https://app.example.com
/callback&scope=openid profile
api:read&state=xyz123&code_challenge=E9Melhoa2OwvFrEMT...&code_challenge_method=S256
HTTP/1.1Host: auth.zentra.local
HTTP/1.1 302 FoundLocation:
https://app.example.com/callback?code=SplxlOBeZQQYbYS6WxSbIA&state=xyz123
```

Negative Case - Invalid Client:

```
GET /connect/authorize?client_id=invalid-client HTTP/1.1Host: auth.zentra.local
HTTP/1.1 400 Bad Request{  "error": "invalid_client",  "error_description": "Client not
found or disabled"}
```

Regression Test: OAuth2_AuthorizeEndpoint_ValidClient_ReturnsCode

Token Endpoint

Positive Case - Authorization Code Exchange:

```
POST /connect/token HTTP/1.1Host: auth.zentra.localContent-Type:
application/x-www-form-urlencodedgrant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA&r
edirect_uri=https://app.example.com/callback&client_id=web-app&client_secret=client-secret&c
ode_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk
HTTP/1.1 200 OKContent-Type: application/json{  "access_token": "eyJhbGciOiJSUzI1NiIs...",
"token_type": "Bearer",  "expires_in": 900,  "refresh_token": "8xLOxBtZp8...",  "scope":
"openid profile api:read"}
```

Regression Test: OAuth2_TokenEndpoint_ValidCode_ReturnsTokens

Refresh Token Flow

Positive Case - Token Refresh:

```
POST /connect/token HTTP/1.1Host: auth.zentra.localContent-Type:
application/x-www-form-urlencodedgrant_type=refresh_token&refresh_token=8xLOxBtZp8...&client
_id=web-app&client_secret=client-secret
HTTP/1.1 200 OK{  "access_token": "eyJhbGciOiJSUzI1NiIs...",  "token_type": "Bearer",
"expires_in": 900,  "refresh_token": "NEWxLOxBtZp8...",  "scope": "openid profile api:read"}
```

Note: New refresh_token issued, previous token invalidated (rotation).

9.2 RFC 7636 - PKCE

S256 Code Challenge Verification

Positive Case - Valid PKCE Verification:

```
// Client computes code_challengeBASE64URL(SHA256(code_verifier))// code_verifier:
dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk// code_challenge: E9Melhoa2OwvFrEMT...
POST /connect/token HTTP/1.1...code_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk
HTTP/1.1 200 OK// Token issued - PKCE verification successful
```

Negative Case - Invalid Code Verifier:

```
POST /connect/token HTTP/1.1...code_verifier=wrong-verifier
HTTP/1.1 400 Bad Request{  "error": "invalid_grant",  "error_description": "PKCE
verification failed"}
```

Regression Test: PKCE_S256_ValidVerifier_ReturnsTokens, PKCE_S256_InvalidVerifier_ReturnsError

9.3 RFC 7009 - Token Revocation

Token Revocation with Idempotency

Positive Case - Successful Revocation:

```
POST /connect/revocation HTTP/1.1Host: auth.zentra.localContent-Type:
application/x-www-form-urlencodedAuthorization: Basic
d2ViLWFwcDpjbGllbnQtc2VjcmV0token=8xLOxBtZp8...&token_type_hint=refresh_token
HTTP/1.1 200 OK// Token revoked successfully
```

Idempotency Verification - Second Revocation Request:

```
POST /connect/revocation HTTP/1.1...same token...
HTTP/1.1 200 OK// Same response - idempotent operation
```

Regression Test: Revocation_ValidToken_Returns200, Revocation_Idempotent_Returns200

9.4 RFC 7662 - Token Introspection

Active Token Introspection

Positive Case - Active Token:

```
POST /connect/introspect HTTP/1.1Host: auth.zentra.localContent-Type:
application/x-www-form-urlencodedAuthorization: Basic
d2ViLWFwcDpjbGllbnQtc2VjcmV0token=eyJhbGciOiJSUzI1NiIs...
HTTP/1.1 200 OKContent-Type: application/json{  "active": true,  "sub": "user@example.com",
"client_id": "web-app",  "exp": 1704067200,  "scope": "openid profile api:read",
"token_type": "Bearer"}
```

Negative Case - Expired/Revoked Token:

```
POST /connect/introspect HTTP/1.1...expired token...
HTTP/1.1 200 OK{  "active": false}
```

Regression Test: Introspect_ActiveToken_ReturnsClaims, Introspect_ExpiredToken_ReturnsActiveFalse

10. STRIDE Threat Model

The following threat model applies STRIDE methodology to identify and mitigate threats across six categories.

10.1 Threat Matrix

Threat	Risk	Mitigation	Residual	Monitoring
Spoofing	Client/user impersonation	Basic Auth, PKCE, RS256	Low	Failed auth logging
Tampering	Token/code modification	JWT signatures, PKCE binding	Low	Signature failure alerts
Repudiation	Denial of actions	Audit trail, correlation IDs	Low	Weekly audit review
Info Disclosure	Token leakage	TLS, hash-at-rest, env secrets	Low	Access log review
DoS	Endpoint flooding	Rate limiting (20/120 req/min)	Medium	Rate limit breach alerts
Elevation	Privilege escalation	Scope enforcement, audience validation	Low	Scope mismatch alerts

10.2 Threat Details

Spoofing (S)

Attack: Attacker impersonates legitimate client or user. Mitigation: Client authentication via Basic Auth, PKCE prevents code interception, RS256 signatures verify token authenticity.

Tampering (T)

Attack: Modification of tokens or authorization codes. Mitigation: JWT signatures detect tampering, PKCE binding ensures code-verifier match.

Repudiation (R)

Attack: User denies performing action. Mitigation: Comprehensive audit trail with correlation IDs enables action tracing.

Information Disclosure (I)

Attack: Unauthorized access to sensitive data. Mitigation: TLS encryption, hashed storage, environment-based secrets.

Denial of Service (D)

Attack: Endpoint flooding to cause outage. Mitigation: Rate limiting (20 req/min per client, 120 req/min per IP). Residual: Distributed attacks require additional infrastructure.

Elevation of Privilege (E)

Attack: Gaining unauthorized permissions. Mitigation: Scope enforcement at API Gateway, audience validation prevents cross-service token use.

11. Residual & Accepted Risk

This section documents risks that remain after mitigation efforts and risks explicitly accepted by the organization.

11.1 Residual Risks

The following risks have been identified with likelihood and impact assessments:

ID	Risk	Likelihood	Impact	Mitigation
RR-001	Private key compromise	Low	Critical	90-day rotation, key vault storage
RR-002	Database compromise with active tokens	Low	High	Encryption at rest, 15-min token TTL
RR-003	Distributed DoS	Medium	Medium	Rate limiting, cloud DDoS protection
RR-004	Insider threat	Low	High	Audit logging, least privilege, separation
RR-005	Dependency vulnerability	Medium	Medium	Weekly scanning, SBOM, rapid patching

11.2 Accepted Risks

The following risks are explicitly accepted with documented justification:

AR-001: No Distributed Rate Limiting

Risk: Single-instance rate limiting may not prevent distributed attacks.

Justification: Current single-instance deployment. Distributed rate limiting via Redis planned for Q2 2026.

Compensating Control: Cloud-level DDoS protection, IP-based rate limiting.

AR-002: No Hardware Security Module (HSM)

Risk: Software key storage more vulnerable than hardware protection.

Justification: 90-day key rotation compensates for software storage risk. HSM procurement scheduled Q2 2026.

Compensating Control: Short key lifetime, key vault with access controls.

AR-003: Self-Signed Certificates in Development

Risk: Certificate validation bypassed in dev environments.

Justification: Development environment isolated from production. Production requires valid CA certificates.

Compensating Control: Environment separation, production hardening checklist.

AR-004: No Real-Time JWT Revocation

Risk: Compromised tokens valid until expiry.

Justification: 15-minute token TTL limits exposure window. Token introspection available for critical scenarios.

Compensating Control: Short TTL, refresh token rotation, rapid key rotation on compromise.

12. Security Monitoring & Detection

Comprehensive monitoring and alerting capabilities for security event detection and response.

12.1 Structured Logging

Serilog implementation with file and database sinks:

Level	Retention	Use Case
Fatal	7 years	System failure, security breach
Error	3 years	Authentication failures, exceptions
Warning	1 year	Rate limit approaches, anomalies
Information	90 days	Normal operations, token issuance
Debug	7 days	Development troubleshooting

12.2 Alert Triggers

Security alerts configured with severity levels:

Alert ID	Trigger	Severity	Response
AUTH-001	5+ failed logins / 5 min	High	Account lockout, notify admin
AUTH-002	Token signature failure	Critical	Log, alert security team
RATE-001	Rate limit exceeded	Medium	Throttle, log, notify
TOKEN-001	Refresh token reuse detected	High	Revoke all tokens, force re-auth
SCOPE-001	Scope abuse attempt	Medium	Log, alert, review
GEO-001	Impossible travel detected	Medium	Require MFA, notify user
CERT-001	Certificate expiry < 30 days	Medium	Alert ops team
CERT-002	Certificate expired	Critical	Emergency rotation

12.3 Token Anomaly Detection

- Token Reuse Detection - Same refresh token used twice indicates potential theft

- Scope Abuse - Requests exceeding granted scopes
- Geographic Anomalies - Login from impossible locations
- Velocity Checks - Unusual token issuance rates

12.4 Rate Limit Monitoring

Rate limit configuration with partition key tracking:

```
// Rate limit configuration{  "client_id": {    "limit": 20,    "window": "1m"  },
"ip_address": {    "limit": 120,    "window": "1m"  }}
```

Breach alerts include partition key for rapid identification of source.

12.5 Key Misuse Detection

- Invalid kid - Token signed with unknown key identifier
- Expired certificate - Token signed with expired signing key
- Algorithm mismatch - Token using unexpected signing algorithm

13. Governance Model

Operational procedures and governance controls for maintaining security posture.

13.1 Key Rotation Schedule

Cryptographic material rotation frequencies:

Key Type	Frequency	Responsible	Procedure
RSA/ECDSA Signing	90 days	Security Engineering	Automated with grace period
Client Secrets	60 days	Operations	Manual rotation with notification
Database Credentials	Quarterly	DBA + Operations	Coordinated rotation
API Keys	30 days	DevOps	Automated regeneration
TLS Certificates	Annual	DevOps	Let's Encrypt or CA renewal

13.2 Dependency Scanning

Vulnerability scanning schedule and tools:

Scan Type	Frequency	Tool	Action on Findings
NuGet vulnerabilities	Weekly	dotnet list package --vulnerable	Patch within 7 days (High/Critical)
Filesystem scan	Weekly	Trivy fs .	Remediate before next release
Container images	Per build	Trivy image	Block deployment if Critical
SBOM generation	Per release	dotnet sbom	Archive for audit

13.3 Testing Policy

Security testing requirements:

Test Suite	Trigger	Pass Rate	Failure Action
Architecture tests	Per PR	100%	Block merge
Security regression	Per PR	100%	Block merge
Integration tests	Per PR + Nightly	95%	Investigate, may override
Penetration tests	Quarterly	All High/Critical remediated	Hold release if needed

13.4 Security Review Process

- Quarterly Security Reviews - Comprehensive control assessment
- Annual Penetration Testing - Third-party security assessment
- Incident Post-Mortems - Root cause analysis for security events
- Threat Model Updates - STRIDE review when architecture changes

14. Maturity Assessment

Evidence-based maturity calibration against actual implementation.

14.1 Recalibrated Claim

Previous marketing claim of L4 - Managed has been recalibrated to L2.5 - Managed based on evidence assessment.

Domain	Level	Evidence
Implementation	L3	Architecture tests, CI gates, code coverage
Verification	L2	Security regression tests, dependency scanning
Operations	L2	Rate limiting, health checks, structured logging
Governance	L2	Key rotation SOP, documented procedures, quarterly reviews

14.2 Gap Analysis

Areas requiring improvement to reach L3:

- Distributed Rate Limiting - Requires Redis implementation (Q2 2026)
- HSM Integration - Hardware key storage procurement (Q2 2026)
- Real-time Revocation - JWT blacklist or short TTL optimization
- Advanced Monitoring - ML-based anomaly detection (roadmap)

14.3 Evidence Documentation

All maturity claims supported by:

- Regression test suite - 100+ security tests
- CI/CD pipeline - Automated security gates
- Audit logs - 90-day retention minimum
- SOP documentation - Key rotation, incident response

15. Conclusion

Zentra v1.1 represents a maturation from marketing claims to evidence-based security documentation. This whitepaper provides auditors, security teams, and procurement professionals with verifiable proof of security controls.

Key Security Attributes

- Protocol Compliant - RFC 6749, 7636, 7009, 7662 with HTTP evidence
- Threat Modeled - Complete STRIDE analysis with mitigations
- Risk Transparent - Documented residual and accepted risks
- Monitored - Comprehensive logging and alerting
- Governed - Key rotation, dependency scanning, testing policy
- Evidence-Based - Maturity claims backed by implementation proof

Audit Readiness

This document provides the foundation for:

- SOC 2 Type II audits - Control evidence and monitoring
- Security questionnaires - RFC compliance with examples
- CISO review - Threat model and risk register
- Procurement evaluation - Transparent security posture

Zentra enables organizations to centralize identity securely with evidence-based confidence.

Zentra

Enterprise Identity Platform

FBT (Future Beyond Tech)

security@fbt.dev | www.fbt.dev