Zentra

Security Whitepaper

Enterprise Identity Platform - Security Architecture & Compliance

Version 1.0 | February 2026

FBT (Future Beyond Tech)

Table of Contents

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

1. Executive Summary

Purpose

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 a comprehensive security overview of Zentra's architecture, protocols, and operational controls for enterprise security teams, auditors, and procurement professionals.

In modern microservices architectures, centralized identity management has become critical. Distributed systems require consistent authentication and authorization mechanisms that can scale without compromising security. Zentra addresses this challenge by providing a vendor-independent, protocol-compliant identity platform that organizations can deploy and control within their own infrastructure.

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)
- Compliance Readiness - Built-in audit logging, access controls, and data protection
- Defense in Depth - Multiple security layers protecting identity infrastructure

This document details Zentra's security architecture, cryptographic implementations, threat mitigations, and operational procedures. It serves as both a technical reference for security engineers and an assurance document for compliance auditors and enterprise procurement teams.

2. Security Philosophy

2.1 Security by Design

Zentra is architected according to established security principles that form the foundation of all design decisions:

Zero Trust Architecture

Zentra operates on the principle that no user, device, or service should be trusted by default. Every access request is authenticated and authorized, regardless of its origin. This approach eliminates implicit trust relationships and ensures consistent security enforcement across all system boundaries.

Principle of Least Privilege

Users and services receive only the minimum permissions necessary to perform their functions. Zentra's scope-based authorization system enforces fine-grained access controls, limiting the blast radius of potential security incidents.

Defense in Depth

Security controls are implemented at multiple layers: network, application, data, and operational. No single control is relied upon exclusively. This layered approach ensures that the compromise of one control does not result in overall system compromise.

Secure Defaults

Zentra ships with secure configurations as the default state. PKCE is enforced for public clients, strong cryptographic algorithms are preferred, and verbose error messages that could aid attackers are suppressed in production environments.

2.2 FBT Engineering Security Standards

Zentra development adheres to FBT's comprehensive security standards, which include:

- No secrets in repository - All credentials, keys, and sensitive configuration are externalized
- Structured logging - Consistent, parseable log formats with security event classification
- Clean Architecture enforcement - Security concerns isolated from business logic
- CI security scanning - Automated vulnerability detection in dependencies and code
- OWASP mitigation - Proactive countermeasures against Top 10 vulnerabilities

3. Identity & Authentication Architecture

3.1 Protocol Standards

Zentra implements industry-standard protocols to ensure interoperability and security:

Protocol	RFC/Standard	Purpose
OAuth 2.0	RFC 6749	Authorization framework for delegated access
OpenID Connect 1.0	OIDC Core	Authentication layer on top of OAuth 2.0
PKCE	RFC 7636	Proof Key for Code Exchange - prevents authorization code interception
Token Introspection	RFC 7662	OAuth token validation endpoint
JWT	RFC 7519	JSON Web Tokens for claims representation
JWK	RFC 7517	JSON Web Key for cryptographic key representation
JWA	RFC 7518	JSON Web Algorithms - RS256, ES256 support

3.2 Supported Flows

Zentra supports the following OAuth 2.0 / OIDC flows, each designed for specific use cases:

Authorization Code Flow

The primary flow for server-side applications. The authorization code is obtained through a user-agent and exchanged for tokens via a back-channel request. This flow supports refresh tokens and is suitable for confidential clients.

Authorization Code + PKCE

Required for public clients (mobile apps, SPAs) and recommended for all clients. PKCE prevents authorization code interception attacks by requiring a code verifier that matches a previously sent code challenge.

Client Credentials Flow

For machine-to-machine authentication where no user is involved. Client applications authenticate using their credentials and receive an access token for service-to-service communication.

Refresh Token Flow

Enables long-lived sessions without storing user credentials. Refresh tokens are rotated on each use and can be revoked independently. This flow supports sliding session windows and device-specific token management.

Device Authorization Flow (Roadmap)

Planned support for device-constrained environments where direct user input is not possible. This flow enables IoT devices and smart TVs to obtain tokens through an out-of-band user authorization process.

4. Token Security Model

4.1 Access Tokens

Access tokens are the primary mechanism for authorization in Zentra. They are designed with security as the paramount concern:

- Cryptographic Signing - All access tokens are signed using RSA (RS256) or ECDSA (ES256) algorithms
- Short-lived - Recommended lifetime of less than 15 minutes minimizes window of compromise
- Audience Restriction - Tokens include intended audience claims preventing cross-service replay
- Scope Embedding - Authorized scopes are cryptographically bound to the token
- Role Claims - User roles and permissions are embedded as signed claims

4.2 Refresh Tokens

Refresh tokens enable long-lived sessions while maintaining security through server-side state management:

- Server-side Storage - Refresh tokens are stored in the SecurityTokens table with metadata
- Token Rotation - A new refresh token is issued on every refresh, invalidating the previous token
- Revocation on Logout - User-initiated logout immediately invalidates all refresh tokens
- Device Isolation - Each device receives independent refresh tokens enabling per-device revocation

The refresh token rotation model provides defense against token theft. If an attacker obtains a refresh token, its use by either the legitimate user or the attacker will invalidate all copies, alerting the system to potential compromise.

4.3 Token Validation

Token validation occurs at multiple points in the request lifecycle:

API Gateway Validation

The API Gateway performs signature verification using the JWKS endpoint. Invalid signatures result in immediate request rejection before reaching backend services.

Resource Server Validation

Resource servers validate audience claims, issuer identification, and token expiry. This ensures tokens are used only by their intended recipients and within their validity period.

Introspection Endpoint

For opaque tokens or additional validation requirements, the introspection endpoint provides real-time token status checking, including active/inactive state and associated metadata.

5. Cryptographic Standards

5.1 Key Management

Zentra implements robust key management practices to protect the cryptographic foundation of the identity platform:

- Asymmetric Signing Keys - RSA or ECDSA key pairs for token signing, with private keys never exposed
- JWKS Endpoint - Public keys are published via a JSON Web Key Set endpoint for consumer verification
- Automated Rotation - Planned support for automated key rotation with grace periods for key propagation
- Environment Separation - Distinct signing keys for development, staging, and production environments

5.2 Algorithm Support

Zentra supports the following cryptographic algorithms for token signing:

Algorithm	Type	Status	Notes
RS256	RSA + SHA-256	Default	Recommended for broad compatibility
ES256	ECDSA + SHA-256	Optional	Smaller signature size, faster verification
HS256	HMAC + SHA-256	Internal Only	For internal service tokens only

5.3 Transport Security

All communications with Zentra are protected via TLS:

- HTTPS Enforcement - All endpoints require TLS encryption
- TLS 1.2+ Minimum - Older TLS versions are rejected
- HSTS Recommended - HTTP Strict Transport Security headers for client protection
- Secure Cipher Suites - Only strong cipher configurations are supported

6. Authorization & Access Control

6.1 Role-Based Access Control (RBAC)

Zentra implements a comprehensive RBAC system for managing user permissions:

- Centralized Role Storage - Role definitions are stored centrally and enforced consistently
- Role Claims in JWT - User roles are embedded as signed claims in access tokens
- API-level Enforcement - Resource servers validate role claims before processing requests
- Hierarchical Roles - Support for role inheritance and permission aggregation

6.2 Scope-Based Authorization

Fine-grained API access is controlled through OAuth 2.0 scopes:

- Granular API Scopes - Each API endpoint can require specific scopes
- Per-Client Allowed Scopes - Clients are restricted to explicitly authorized scope sets
- Gateway-level Enforcement - Scope validation at the API Gateway prevents unauthorized access
- Dynamic Scope Requests - Users can grant partial scope approvals during consent

6.3 Client-Level Restrictions

OAuth 2.0 client configurations enforce security boundaries:

- Redirect URI Validation - Authorization codes are bound to registered redirect URIs
- PKCE Enforcement - Public clients must use PKCE; confidential clients are strongly encouraged
- Grant Type Restrictions - Clients are limited to explicitly permitted OAuth flows
- Client Authentication - Confidential clients authenticate using client secrets or certificates

7. Multi-Factor Authentication

Zentra provides comprehensive MFA capabilities to protect against credential-based attacks:

7.1 Supported Authentication Factors

Factor	Type	Implementation
Email OTP	Possession	Time-limited codes sent via email
SMS OTP	Possession	Time-limited codes sent via SMS
Authenticator App	Possession	TOTP (RFC 6238) compatible with Google/Microsoft Authenticator
LDAP + 2FA	Knowledge + Possession	Integration with enterprise LDAP with additional factor

7.2 Security Measures

MFA implementations include protective measures against abuse:

- Retry Limits - Failed MFA attempts trigger progressive delays
- Account Lockout - Repeated failures result in temporary account suspension
- Configurable Policy - Administrators can require MFA per user group or application
- Backup Codes - Single-use recovery codes for account recovery scenarios
- Device Trust - Remembered devices can be exempted from repeated MFA challenges

8. Audit & Monitoring

8.1 Audit Logging

Zentra maintains comprehensive audit trails for all security-relevant events. The following events are logged with detailed metadata:

- Login Events - Successful and failed authentication attempts
- Token Lifecycle - Token issuance, refresh, and revocation events
- Administrative Actions - Role assignments, client creation, configuration changes
- MFA Events - Factor enrollment, verification attempts, and failures
- Consent Events - User approvals and denials of scope requests

Each audit record includes the following fields:

Field	Description
Actor	User or service principal that performed the action
Timestamp	UTC timestamp with millisecond precision
Event Type	Categorized security event classification
IP Address	Source IP (if configured for collection)
User Agent	Client application identification
Old/New Values	Before/after state for modification events
Correlation ID	Request tracing identifier

8.2 Observability

Zentra provides operational visibility through structured logging and health monitoring:

- Structured Logging - JSON-formatted logs compatible with SIEM systems
- Correlation IDs - Request tracing across distributed components
- Health Endpoints - Planned availability of liveness and readiness probes
- OpenTelemetry Integration - Planned support for distributed tracing
- Metrics Export - Performance and security metrics for monitoring systems

9. OWASP & Threat Mitigation

Zentra implements comprehensive countermeasures against the OWASP Top 10 and other common attack vectors:

OWASP Risk	Mitigation Strategy
Injection	Parameterized queries via Entity Framework Core; input validation
Broken Authentication	PKCE enforcement, token rotation, secure session management
Sensitive Data Exposure	TLS enforcement, encrypted cookies, secure headers
XML External Entities	XML parser hardening, DTD disabling
Broken Access Control	RBAC enforcement, scope validation, claim verification
Security Misconfiguration	Secure defaults, minimal feature exposure, hardening guides
XSS	Razor automatic encoding, Content Security Policy headers
Insecure Deserialization	Type constraints, input validation, safe parsers
Using Components with Vulnerabilities	Dependency scanning, SBOM generation, update policies

OWASP Risk	Mitigation Strategy
Insufficient Logging	Comprehensive audit logging, security event monitoring

Additional Threat Mitigations

CSRF Protection

Anti-forgery tokens are required for state-changing operations. SameSite cookie policies provide additional defense against cross-site request forgery.

Open Redirect Prevention

Redirect URI validation ensures authorization codes are only sent to pre-registered locations. Local URL validation prevents malicious redirects after authentication.

Brute Force Protection

Account lockout policies progressively delay authentication attempts after failures. Rate limiting on token endpoints prevents automated credential stuffing attacks.

Replay Attack Prevention

Authorization codes are single-use and time-limited. Nonce validation in OIDC flows prevents token replay attacks.

Token Theft Mitigation

Short access token lifetimes limit exposure windows. Refresh token rotation detects and invalidates stolen tokens. Binding tokens to client instances where possible.

10. Zero Trust Model

Zentra implements a Zero Trust security model that assumes breach and verifies every request:

10.1 Core Principles

Never Trust, Always Verify

Every access request is authenticated and authorized, regardless of source. Internal service communication requires the same validation as external requests. No implicit trust is granted based on network location.

Assume Breach

Zentra is designed with the assumption that attackers may already have some level of access. Defense in depth, least privilege, and comprehensive logging provide detection and containment capabilities.

Verify Explicitly

Authentication and authorization decisions are based on all available data points: user identity, device health, service identity, data classification, and anomaly detection.

10.2 Implementation

- Service-to-Service JWT Validation - All inter-service calls include validated JWT tokens
- Explicit Scope Enforcement - Every API call validates required scopes
- Tenant Claim Validation - Multi-tenant deployments validate tenant context (roadmap)
- No Implicit Trust - Services do not trust each other based on network proximity
- Continuous Validation - Sessions are re-evaluated throughout their lifetime

11. Data Protection

11.1 Encryption in Transit

All data transmission is protected using industry-standard encryption:

- HTTPS Enforcement - All endpoints require TLS 1.2 or higher
- Secure Cookies - Authentication cookies use Secure and HttpOnly flags
- SameSite Policies - Cookie SameSite attributes prevent CSRF attacks
- HSTS Headers - HTTP Strict Transport Security for browser protection

11.2 Encryption at Rest

Data storage protections include:

- Token Encryption - Planned encryption of sensitive token data at rest
- Database Encryption - Support for transparent database encryption via provider
- Secrets Management - Credentials stored in environment variables or vault solutions
- Key Separation - Encryption keys managed independently from data

11.3 Data Minimization

Zentra follows data minimization principles:

- Minimal Claims - Tokens contain only necessary identity claims
- Scope-Based Inclusion - Claims are included based on authorized scopes
- PII Limitation - Personal data exposure is minimized in logs and tokens
- Retention Policies - Configurable data retention with automatic purging

12. Secrets Management

Zentra follows FBT's secrets management hierarchy for credential protection:

12.1 Priority Hierarchy

Priority	Method	Use Case
1 (Highest)	Managed Identity	Cloud-native workload identity (Azure AD, AWS IAM)
2	Key Vault	Centralized secret storage with access controls
3	Environment Variables	Container and VM deployments
4 (Never)	Configuration Files	Explicitly prohibited - never commit secrets

12.2 Secret Types

The following secrets require protection according to the hierarchy above:

- Database Connection Strings - Including credentials and connection parameters
- Signing Keys - Private keys for JWT token signing
- API Keys - External service integration credentials
- Client Secrets - OAuth 2.0 confidential client credentials
- Encryption Keys - Data-at-rest encryption keys

12.3 CI/CD Security

Continuous integration pipelines include security scanning:

- Secret Scanning - Automated detection of committed credentials
- Dependency Scanning - Vulnerability detection in third-party libraries
- Static Analysis - Code security analysis during build

- Container Scanning - Vulnerability assessment of deployment images

13. Infrastructure Security

Zentra deployment recommendations include layered infrastructure protections:

13.1 Network Security

- Reverse Proxy - Nginx or Azure Application Gateway for TLS termination
- Web Application Firewall - WAF rules for common attack patterns
- Rate Limiting - Mandatory rate limiting on token endpoints
- DDoS Protection - Cloud-level protection against volumetric attacks
- Network Segmentation - Isolate identity services in secure network zones

13.2 Deployment Hardening

- Minimal Attack Surface - Disable unused features and endpoints
- Security Headers - HSTS, CSP, X-Frame-Options, X-Content-Type-Options
- Container Security - Non-root execution, read-only filesystems
- Resource Limits - CPU and memory constraints for DoS protection

13.3 High Availability

Production deployments should implement redundancy for availability and security:

- Multi-Instance Deployment - Multiple Zentra instances behind load balancer
- Database Clustering - Replicated database for failover protection
- Health Monitoring - Automated health checks and failure detection
- Backup Procedures - Regular encrypted backups with tested recovery

14. Compliance Readiness

Zentra is designed to support enterprise compliance requirements:

Standard	Capability	Implementation
SOC 2	Security Controls	Audit logging, access control, change management
GDPR	Data Protection	Data export, deletion capabilities, consent management
HIPAA	Healthcare Security	Audit logging, encryption, access controls
ISO 27001	ISMS	Policy enforcement, risk management, security procedures
PCI DSS	Payment Security	Secure authentication (no card data storage)
CCPA	Privacy Rights	Data disclosure, deletion, opt-out mechanisms

14.1 Audit Support

Zentra facilitates compliance audits through:

- Comprehensive Logging - All security events logged with tamper-evident records
- Access Reports - User and administrator access reporting capabilities
- Configuration Documentation - Security configuration baselines and drift detection
- Evidence Collection - Automated evidence gathering for audit requests

15. Operational Security Controls

Ongoing security operations are supported by automated and manual controls:

15.1 Continuous Security

- CI/CD Security Scanning - Automated vulnerability detection in build pipeline
- Dependency Monitoring - Continuous tracking of third-party vulnerabilities
- SBOM Generation - Software Bill of Materials for supply chain transparency
- Container Scanning - Vulnerability assessment of deployment images
- Penetration Testing - Recommended annual third-party security assessments

15.2 Vulnerability Management

Security vulnerabilities are managed through a structured process:

- Discovery - Automated scanning, researcher reports, threat intelligence
- Assessment - Severity classification and impact analysis
- Remediation - Patch development and deployment procedures
- Disclosure - Responsible disclosure coordination and communication

15.3 Change Management

Security-related changes follow controlled processes:

- Security Review - All changes reviewed for security implications
- Testing - Security testing in staging environments
- Rollback Plans - Contingency procedures for security-related changes
- Documentation - Security configuration and change records

16. Secure Development Lifecycle

Zentra development follows FBT's secure development lifecycle, integrating security at every phase:

16.1 Development Phases

1. Architecture Design

Security requirements are defined alongside functional requirements. Threat modeling identifies potential attack vectors and informs design decisions. Security architecture patterns are selected and documented.

2. Threat Modeling

STRIDE methodology is applied to identify threats across six categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Mitigations are designed for each identified threat.

3. Code Implementation

Secure coding standards are enforced through code review and automated analysis. Input validation, output encoding, and secure API usage are mandatory. Security-sensitive code requires additional review.

4. Security Review

Dedicated security review of architecture and implementation. Penetration testing of new features. Review of security test coverage and effectiveness.

5. Automated Testing

Security tests are integrated into the CI/CD pipeline. Static analysis, dependency scanning, and dynamic testing provide continuous security validation.

6. Hardening

Production configurations are hardened according to security baselines. Unnecessary features are disabled. Security headers and controls are verified.

7. Production Monitoring

Security events are monitored in production. Anomaly detection identifies potential attacks. Incident response procedures are maintained and tested.

17. Incident Response Model

Zentra includes capabilities to support security incident response:

17.1 Log Retention

Security logs are retained according to organizational policies and compliance requirements. Recommended minimum retention periods:

- Authentication Events - 1 year minimum
- Administrative Actions - 3 years minimum
- Security Alerts - 7 years or per compliance requirements

17.2 Audit Trail Query

Security teams can query audit trails using multiple dimensions:

- User Activity - All actions by specific user accounts
- Time Range - Events within specified time windows
- Event Type - Specific security event categories
- IP Address - Activity from specific network sources
- Resource - Access to specific protected resources

17.3 Emergency Procedures

Token Revocation

In the event of suspected token compromise, administrators can immediately revoke all tokens for affected users or clients. Revocation is effective within seconds across all services.

Key Rotation

Emergency key rotation procedures allow rapid replacement of compromised signing keys. Grace periods ensure service continuity during key propagation.

User Lockout

Administrators can immediately disable user accounts in response to security incidents. Lockout is effective across all connected applications.

18. Roadmap to Enterprise Hardened

Zentra's development roadmap includes planned enhancements for enterprise security requirements:

18.1 Near-Term (Current - 6 Months)

- Rate Limiting - Configurable rate limits per client and endpoint
- Token Encryption at Rest - Encrypted storage of sensitive token data
- Health Endpoints - Standardized liveness and readiness probes
- Structured Logging - Enhanced JSON logging with correlation IDs

18.2 Medium-Term (6-12 Months)

- Distributed Tracing - OpenTelemetry integration for request tracing
- Automated Key Rotation - Scheduled key rotation with zero downtime
- Advanced MFA - Biometric and hardware token support
- Risk-Based Authentication - Contextual authentication challenges

18.3 Long-Term (12+ Months)

- Federation Support - SAML 2.0 and WS-Federation integration
- Adaptive Authentication - Machine learning-based anomaly detection
- Identity Proofing - Document verification and biometric matching
- Compliance Automation - Automated compliance evidence collection

19. Security Contact & Disclosure Policy

FBT is committed to responsible security disclosure and collaboration with the security community.

19.1 Reporting Vulnerabilities

Security researchers and users can report vulnerabilities through the following channels:

- Email: security@fbt.dev
- PGP Key: Available at https://fbt.dev/security/pgp-key
- Response Time: Initial acknowledgment within 48 hours

19.2 Disclosure Policy

FBT follows coordinated disclosure practices:

- Private Notification - Vulnerability details kept confidential during remediation
- Fix Timeline - Target 90 days for critical vulnerabilities
- Public Disclosure - Coordinated release after fix availability
- Credit - Researchers acknowledged with permission

19.3 Security Advisories

Security advisories are published for vulnerabilities affecting Zentra. Subscribers can receive notifications through:

- Security Mailing List - security-announce@fbt.dev
- GitHub Security Advisories - github.com/fbt/zentra/security
- RSS Feed - fbt.dev/security/advisories.xml

20. Conclusion

Zentra represents a comprehensive approach to enterprise identity security. By implementing industry-standard protocols, following security best practices, and maintaining a security-first development culture, Zentra provides organizations with a robust foundation for securing their applications and APIs.

Key Security Attributes

- Protocol Compliant - Full OAuth 2.0 and OpenID Connect implementation
- Secure by Design - Security principles embedded in architecture and development
- Architecturally Isolated - Self-hosted deployment maintains organizational control
- Enterprise Ready - Scales to meet demanding enterprise requirements
- Vendor Independent - Open standards prevent vendor lock-in
- Transparent Pricing - No per-user fees enabling unlimited growth

Organizations deploying Zentra gain a security-focused identity platform that protects their users, data, and services while maintaining the flexibility to adapt to evolving security requirements.

Zentra enables organizations to centralize identity securely while maintaining architectural control and compliance readiness.

Zentra

Enterprise Identity Platform

FBT (Future Beyond Tech)

contact@fbt.dev | www.fbt.dev