

# Readme

SECURAA Security Documentation

## SECURAA Secure Software Development Lifecycle (SDLC) Documentation

### Overview

This directory contains comprehensive documentation of SECURAA's Secure Software Development Lifecycle (SDLC). These documents demonstrate our commitment to security-first development practices and provide transparency to our customers about how we build and maintain secure software.

### Document Suite

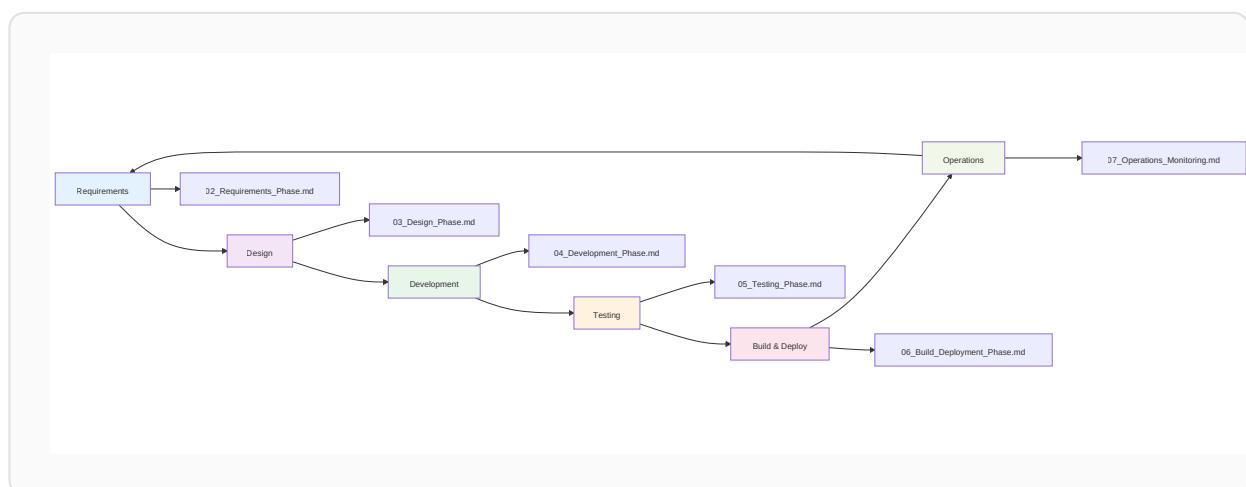
#### Core SDLC Documents

Document	Description	Status
<a href="#">01_SDLC_Overview.md</a>	Executive summary, principles, technology stack, and process overview	 Complete
<a href="#">02_Requirements_Phase.md</a>	Security requirements gathering, threat modeling, risk assessment	 Complete
<a href="#">03_Design_Phase.md</a>	Secure architecture design, security controls, infrastructure design	 Complete
<a href="#">04_Development_Phase.md</a>		 In Progress

Document	Description	Status
	Secure coding, Git workflow on AWS CodeCommit, code review	
<a href="#">05_Testing_Phase.md</a>	Security testing, SAST, DAST, penetration testing	Planned
<a href="#">06_Build_Deployment_Phase.md</a>	AWS CodeBuild, ECR, container security, IAM, deployment	Planned
<a href="#">07_Operations_Monitoring.md</a>	Production operations, monitoring, incident response	Planned
<a href="#">06_CI_CD_Security_Pipeline.md</a>	Automated security in CI/CD, security gates, scan integration	Planned

## Quick Navigation

### By Development Phase



### By Security Activity

Security Activity	Relevant Documents
Threat Modeling	<a href="#">02_Requirements_Phase.md</a>
Architecture Security	<a href="#">03_Design_Phase.md</a>

Security Activity	Relevant Documents
Secure Coding	<a href="#">04_Development_Phase.md</a>
Code Review	<a href="#">04_Development_Phase.md</a>
SAST/DAST	<a href="#">05_Testing_Phase.md</a>
Container Security	<a href="#">06_Build_Deployment_Phase.md</a>
Security Monitoring	<a href="#">07_Operations_Monitoring.md</a>
CI/CD Security	<a href="#">08_CI_CD_Security_Pipeline.md</a>

## By AWS Service

AWS Service	Relevant Documents
AWS CodeCommit	<a href="#">04_Development_Phase.md</a>
AWS CodeBuild	<a href="#">06_Build_Deployment_Phase.md</a> , <a href="#">08_CI_CD_Security_Pipeline.md</a>
Amazon ECR	<a href="#">06_Build_Deployment_Phase.md</a>
AWS IAM	<a href="#">03_Design_Phase.md</a> , <a href="#">06_Build_Deployment_Phase.md</a>
AWS Secrets Manager	<a href="#">03_Design_Phase.md</a> , <a href="#">06_Build_Deployment_Phase.md</a>
Amazon CloudWatch	<a href="#">07_Operations_Monitoring.md</a>

---

## Document Structure

---

Each phase document follows a consistent structure:

1. **Document Control** - Version, date, classification
2. **Phase Overview** - Objectives, duration, key activities
3. **Process Details** - Step-by-step processes with examples
4. **Security Controls** - Specific security measures
5. **Tools & Templates** - Tools used and available templates

6. **Checklist** - Phase completion checklist
  7. **Exit Criteria** - Requirements to proceed to next phase
- 

## SDLC Process Summary

---

### Phase 1: Requirements & Planning

**Duration:** 1-2 weeks | **Document:** [02\\_Requirements\\_Phase.md](#)

- Gather security requirements
- Conduct threat modeling (STRIDE)
- Perform risk assessment
- Define security acceptance criteria

**Key Deliverables:** - Security requirements document - Threat model - Risk assessment - Security acceptance criteria

---

### Phase 2: Secure Design

**Duration:** 1-3 weeks | **Document:** [03\\_Design\\_Phase.md](#)

- Design secure architecture
- Define security controls
- Design data protection
- Design API security
- Design infrastructure security

**Key Deliverables:** - Architecture diagrams - Security design document - Database security design - API security specifications - Infrastructure security design

---

### Phase 3: Secure Development

**Duration:** Ongoing | **Document:** [04\\_Development\\_Phase.md](#)

- Follow secure coding standards
- Use Git workflow on AWS CodeCommit
- Implement security controls

- Conduct code reviews
- Run pre-commit security checks

**Key Deliverables:** - Secure source code - Code review approvals - Unit tests with security focus - Development documentation

---

## Phase 4: Security Testing

**Duration:** 1-2 weeks | **Document:** [05\\_Testing\\_Phase.md](#)

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Dependency vulnerability scanning
- Security acceptance testing
- Penetration testing

**Key Deliverables:** - SAST reports - DAST reports - Vulnerability scan results - Security test results - Penetration test report

---

## Phase 5: Build & Deployment

**Duration:** Continuous | **Document:** [06\\_Build\\_Deployment\\_Phase.md](#)

- AWS CodeBuild pipeline execution
- Docker image building and scanning
- Push to Amazon ECR
- Security gate validation
- Deployment automation

**Key Deliverables:** - Secure Docker images - Build artifacts (RPM packages) - Deployment packages - Deployment documentation

---

## Phase 6: Operations & Monitoring

**Duration:** Continuous | **Document:** [07\\_Operations\\_Monitoring.md](#)

- Security monitoring
- Log analysis

- Incident detection and response
- Performance monitoring
- Security patching

**Key Deliverables:** - Monitoring dashboards - Security alerts - Incident reports - Performance metrics - Patch management records

---

## Technology Stack

---

### Development

- **Backend:** Go 1.17+ (530+ microservices)
- **Frontend:** React 18.2, Redux
- **Scripting:** Python 3.8+, Bash
- **Database:** MongoDB 7.0, PostgreSQL

### AWS Infrastructure

- **Version Control:** AWS CodeCommit
- **CI/CD:** AWS CodeBuild
- **Container Registry:** Amazon ECR
- **Compute:** Amazon EC2
- **Security:** AWS IAM, AWS Secrets Manager, AWS KMS
- **Monitoring:** Amazon CloudWatch
- **Networking:** AWS VPC, ALB, Security Groups

### Security Tools

- **SAST:** GoSec, ESLint Security Plugin, Bandit
  - **Dependency Scanning:** Nancy, npm audit, govulncheck
  - **DAST:** OWASP ZAP
  - **Container Scanning:** Docker Scan, Trivy, AWS ECR Scanning
  - **Secrets Detection:** git-secrets, TruffleHog
-

# Repository Structure

---

```
SECURAA Platform Repositories:  
├── build_securaa/          # Build automation, CI/CD configs  
├── zona_services/          # Core microservices (530+ services)  
├── securaa/                # Main application backend  
├── securaa_lib/            # Shared security library  
├── zona_batch/             # Background batch processing  
├── integrations/           # Third-party integrations (722+)  
├── zonareact/              # Frontend React application  
├── securaa_db/              # Database schemas and migrations  
└── securaa_pylib/          # Python utility libraries
```

---

## Security Principles

---

### 1. Security by Design

Security is built into every phase, not added as an afterthought.

### 2. Defense in Depth

Multiple layers of security controls protect against various threats.

### 3. Least Privilege

Minimum necessary permissions for users and services.

### 4. Continuous Security

Ongoing security assessment and improvement.

### 5. Shift Left Security

Security testing as early as possible in development.

### 6. Automation First

Automated security checks ensure consistency.

---

# Compliance Alignment

Our SDLC supports compliance with:

Framework	Key Controls	Documentation
<b>SOC 2 Type II</b>	CC6.1, CC6.6, CC6.7, CC7.2, CC8.1	All SDLC phases
<b>ISO 27001:2022</b>	A.8.1, A.8.3, A.14.2, A.14.2.5, A.18.1.3	All SDLC phases
<b>GDPR</b>	Data protection by design, security of processing	Requirements, Design phases
<b>OWASP Top 10</b>	A01-A10 coverage	Testing, Development phases

## Key Metrics & KPIs

### Security Metrics

Metric	Target	Tracking
<b>Critical Vulnerabilities</b>	0	Continuous
<b>High Vulnerabilities</b>	< 5	Weekly
<b>Mean Time to Remediation (Critical)</b>	< 24 hours	Per incident
<b>Security Test Coverage</b>	> 80%	Monthly
<b>Failed Security Scans</b>	< 5%	Per build

### Development Metrics

Metric	Target	Tracking
<b>Build Success Rate</b>	> 95%	Daily
<b>Code Review Completion Time</b>	< 48 hours	Per PR

Metric	Target	Tracking
Deployment Frequency	Multiple per week	Weekly
Change Failure Rate	< 10%	Monthly

---

## Related Documentation

---

### Additional Security Documents

- [\*\*SECURAA\\_SECURE\\_CODING\\_POLICY.md\*\*](#) - Comprehensive secure coding standards
  - [\*\*SECURAA\\_CODE\\_ANALYSIS\\_REPORT.md\*\*](#) - Security code analysis findings
  - [\*\*SECURAA\\_INFORMATION\\_SECURITY\\_POLICIES\\_AND\\_PROCEDURES.md\*\*](#) - Information security policies
  - [\*\*SECURAA\\_INFORMATION\\_SECURITY\\_RISK\\_ASSESSMENT\\_PROCESS.md\*\*](#) - Risk assessment framework
  - [\*\*SECURAA\\_CUSTOMER\\_SECURITY\\_DOCUMENTATION.md\*\*](#) - Customer-facing security overview
- 

## How to Use This Documentation

---

### For Developers

1. Review [\*\*04\\_Development\\_Phase.md\*\*](#) for secure coding practices
2. Follow Git workflow on AWS CodeCommit
3. Use provided code review checklists
4. Run security scans before committing

### For Security Team

1. Use [\*\*02\\_Requirements\\_Phase.md\*\*](#) for threat modeling
2. Conduct design reviews using [\*\*03\\_Design\\_Phase.md\*\*](#)
3. Define security test cases from [\*\*05\\_Testing\\_Phase.md\*\*](#)
4. Monitor using [\*\*07\\_Operations\\_Monitoring.md\*\*](#)

## For DevOps Team

1. Configure CI/CD using [08\\_CI\\_CD\\_Security\\_Pipeline.md](#)
2. Manage deployments with [06\\_Build\\_Deployment\\_Phase.md](#)
3. Set up monitoring from [07\\_Operations\\_Monitoring.md](#)

## For Product Managers

1. Start with [01\\_SDLC\\_Overview.md](#) for process overview
2. Define security requirements using [02\\_Requirements\\_Phase.md](#)
3. Review security acceptance criteria before release

## For Customers

1. Read [01\\_SDLC\\_Overview.md](#) for our security commitment
  2. Review compliance alignment for your requirements
  3. Understand our security testing approach
- 

# Document Maintenance

---

## Version Control

- All documents are version controlled in Git
- Major updates require security team review
- Quarterly review cycle

## Update Process

1. Propose changes via pull request
2. Security team review
3. Engineering leadership approval
4. Merge and publish

## Feedback

We welcome feedback on these documents: - **Email:** [sdlc-feedback@securaa.com](mailto:sdlc-feedback@securaa.com) - **Slack:** #security-sdlc - **JIRA:** Create ticket in SECURITY project

---

# Document History

---

Version	Date	Author	Changes
1.0	October 2024	Security Team	Initial SDLC documentation
2.0	November 13, 2025	Security & Engineering	Comprehensive update with AWS infrastructure, multi-document structure

---

## Contact Information

---

**Document Owners:** - **Security Team:** security@securaa.com - **Engineering Team:** engineering@securaa.com - **DevOps Team:** devops@securaa.com

**For Security Concerns:** - **Email:** security@securaa.com - **Emergency:** security-oncall@securaa.com - **Bug Bounty:** bugbounty@securaa.com

---

## License & Confidentiality

---

These documents are: - **Confidential:** Intended for customers under NDA - **Customer-Facing:** Can be shared with customers - **Not Public:** Not for public distribution - **Proprietary:** © 2025 SECURAA. All rights reserved.

---

## Next Steps

---

- 1. Start Here:** [01\\_SDLC\\_Overview.md](#)
  - 2. Learn Process:** Follow documents 02-08 in sequence
  - 3. Apply Knowledge:** Use checklists and templates in your projects
  - 4. Continuous Improvement:** Provide feedback and suggestions
-

*Last Updated: November 13, 2025*

*Document Suite Version: 2.0*

*Maintained by: SECURAA Security & Engineering Teams*