CSrental AI - Security Guide

Security Overview

CSrental AI implements enterprise-grade security measures to protect sensitive company data and ensure secure access to AI capabilities.

Authentication & Authorization

Two-Factor Authentication (2FA)

Implementation

- **TOTP-based**: Compatible with Google Authenticator, Authy, etc.
- **Backup Codes**: 10 single-use recovery codes
- **Mandatory for Admins**: Required for admin role access

Setup Process

1. User navigates to 2FA setup
2. QR code generated with secret
3. User scans with authenticator app
4. Verification with 6-digit code
5. Backup codes generated and displayed

Security Features

- **Time-based codes**: 30-second validity window
- **Rate limiting**: Max 5 attempts per 15 minutes
- **Secure storage**: Encrypted secrets in database
- **Recovery mechanism**: Backup codes for account recovery

Role-Based Access Control (RBAC)

User Roles

- **USER**: Basic chat access, document upload
- **ADMIN**: Document approval, user management
- **SUPER_ADMIN**: Full system access, security settings

Permission Matrix

Feature	USER	ADMIN	SUPER_ADMIN
Chat (CeeS/ChriS)			
Document Upload			
Document Approval			
User Management			
System Config			
Audit Logs			

Network Security

IP Whitelisting

Configuration

```
ALLOWED_IPS=192.168.1.0/24,10.0.0.0/8,office.ip.address
```

Implementation

- **Middleware-level**: Blocks requests at application entry
- **Configurable**: Environment-based IP list
- **Wildcard support**: Allow all with '*' (development only)
- **Logging**: All blocked attempts logged

Best Practices

- Use specific IP ranges, not broad subnets
- Regular review of allowed IPs
- VPN integration for remote access
- Emergency access procedures documented

Rate Limiting

Configuration by Endpoint Type

```
export const rateLimitConfigs = {
  api: { limit: 100, windowMs: 15 * 60 * 1000 },     // 100/15min
  auth: { limit: 5, windowMs: 15 * 60 * 1000 },      // 5/15min
  upload: { limit: 10, windowMs: 60 * 60 * 1000 },   // 10/hour
  chat: { limit: 50, windowMs: 15 * 60 * 1000 },     // 50/15min
}
```

Features

- **Per-IP tracking**: Individual limits per client IP
- **Redis support**: Scalable with Redis backend

- **Graceful degradation**: Fallback to memory store
- **Custom headers**: Rate limit info in responses

Data Security

Encryption

Data at Rest

- **Database**: PostgreSQL with encryption at rest
- **File Storage**: Supabase Storage with encryption
- **Secrets**: Environment variables encrypted
- **Backups**: Encrypted backup storage

Data in Transit

- **HTTPS**: TLS 1.3 for all communications
- **API calls**: Encrypted OpenAI API requests
- **Database**: SSL connections required
- **Internal**: Encrypted service-to-service communication

Data Classification

Sensitive Data

- **User credentials**: Hashed passwords, 2FA secrets
- **Personal information**: Email addresses, names
- **Business data**: Uploaded documents, chat history
- **System data**: API keys, configuration

Data Handling

- **Minimal collection**: Only necessary data collected
- **Purpose limitation**: Data used only for intended purpose
- **Retention limits**: Automatic cleanup of old data
- **Access logging**: All data access logged

Application Security

Input Validation

File Uploads

```
// File type validation
const allowedTypes = ['application/pdf', 'text/plain', 'text/markdown']
if (!allowedTypes.includes(file.type)) {
  throw new Error('File type not allowed')
}

// File size validation
if (file.size > MAX_FILE_SIZE) {
  throw new Error('File too large')
}
```

API Inputs

- **Schema validation**: Zod schemas for all inputs
- **Sanitization**: XSS prevention
- **SQL injection**: Parameterized queries only
- **Command injection**: No shell command execution

Security Headers

Implemented Headers

```
{
  'X-Frame-Options': 'DENY',
  'X-Content-Type-Options': 'nosniff',
  'Referrer-Policy': 'strict-origin-when-cross-origin',
  'X-XSS-Protection': '1; mode=block',
  'Permissions-Policy': 'camera=(), microphone=(), geolocation=()'
}
```

Content Security Policy (CSP)

```
default-src 'self';
script-src 'self' 'unsafe-inline';
style-src 'self' 'unsafe-inline';
img-src 'self' data: https:;
connect-src 'self' https://api.openai.com https://*.supabase.co;
```

Audit & Monitoring

Audit Logging

Logged Events

- **Authentication**: Login, logout, 2FA events
- **Authorization**: Permission checks, role changes
- **Data access**: Document views, downloads
- **Administrative**: User management, system changes
- **Security**: Failed attempts, suspicious activity

Log Structure

```
interface AuditLog {
  id: string
  userId?: string
  action: string
  resource?: string
  ipAddress?: string
  userAgent?: string
  metadata?: Record<string, any>
  severity: 'DEBUG' | 'INFO' | 'WARN' | 'ERROR' | 'CRITICAL'
  createdAt: string
}
```

Security Monitoring

Real-time Alerts

- **Failed login attempts**: > 5 failures in 15 minutes
- **Rate limit violations**: Repeated limit breaches
- **Unauthorized access**: Admin area access attempts
- **Suspicious patterns**: Unusual user behavior

Monitoring Dashboards

- **Security events**: Real-time security event feed
- **User activity**: Login patterns, feature usage
- **System health**: Performance and availability
- **Threat detection**: Automated threat identification

Incident Response

Security Incident Classification

Severity Levels

1. **Critical**: Data breach, system compromise
2. **High**: Unauthorized access, service disruption
3. **Medium**: Security policy violation, suspicious activity
4. **Low**: Failed login attempts, minor policy breach

Response Procedures

Immediate Response (0-15 minutes)

1. **Assess impact**: Determine scope and severity
2. **Contain threat**: Block malicious IPs, disable accounts
3. **Notify team**: Alert security team and management
4. **Document**: Record all actions taken

Investigation (15 minutes - 4 hours)

1. **Collect evidence**: Gather logs and system data
2. **Analyze impact**: Determine what was affected
3. **Identify cause**: Root cause analysis
4. **Plan remediation**: Develop fix strategy

Recovery (4-24 hours)

1. **Implement fixes**: Apply security patches
2. **Restore services**: Bring systems back online
3. **Verify security**: Confirm threats eliminated
4. **Monitor closely**: Enhanced monitoring period

Post-Incident (24+ hours)

1. **Lessons learned**: Document what happened
2. **Process improvement**: Update procedures
3. **Training**: Additional security training
4. **Communication**: Stakeholder notification

Compliance

GDPR Compliance

Data Subject Rights

- **Access**: Users can view their data
- **Rectification**: Users can correct their data
- **Erasure**: Users can request data deletion
- **Portability**: Users can export their data

Privacy by Design

- **Data minimization**: Collect only necessary data
- **Purpose limitation**: Use data only for stated purpose
- **Storage limitation**: Delete data when no longer needed
- **Accuracy**: Keep data accurate and up-to-date

Security Standards

ISO 27001 Alignment

- **Information security management**: Documented policies
- **Risk assessment**: Regular security risk reviews
- **Access control**: Principle of least privilege
- **Incident management**: Formal incident response

SOC 2 Type II Considerations

- **Security**: Logical and physical access controls
- **Availability**: System uptime and performance
- **Processing integrity**: Complete and accurate processing
- **Confidentiality**: Protection of confidential information

Security Testing

Automated Testing

Static Analysis

```
# Security linting
npm run lint:security

# Dependency scanning
npm audit

# Code quality
npm run test:security
```

Dynamic Testing

- **Penetration testing**: Quarterly external testing
- **Vulnerability scanning**: Weekly automated scans
- **Security monitoring**: Continuous threat detection
- **Compliance audits**: Annual compliance reviews

Manual Testing

Security Reviews

- **Code reviews**: Security-focused code reviews
- **Architecture reviews**: Security architecture validation
- **Configuration reviews**: Security settings verification
- **Process reviews**: Security procedure validation

Security Maintenance

Regular Tasks

Daily

- [ ] Review security alerts
- [ ] Check failed login attempts
- [ ] Monitor system performance
- [ ] Verify backup completion

Weekly

- [ ] Security log analysis
- [ ] Vulnerability scan review
- [ ] Access review (new/removed users)
- [ ] Security metric reporting

Monthly

- [ ] Security policy review
- [ ] Incident response testing
- [ ] Security training updates
- [ ] Compliance checklist review

Quarterly

- [ ] Penetration testing
- [ ] Security architecture review
- [ ] Disaster recovery testing
- [ ] Security awareness training

Emergency Procedures

Security Breach Response

1. **Immediate isolation**: Disconnect affected systems
2. **Evidence preservation**: Secure logs and data
3. **Stakeholder notification**: Inform management
4. **External support**: Engage security experts if needed

Account Compromise

1. **Disable account**: Immediately suspend access
2. **Reset credentials**: Force password reset
3. **Review activity**: Check for unauthorized actions
4. **Monitor closely**: Enhanced monitoring for account

System Compromise

1. **Isolate system**: Disconnect from network
2. **Preserve evidence**: Create system images
3. **Rebuild system**: Clean installation
4. **Restore from backup**: Use verified clean backup

For security incidents or questions, contact: security@csrental.nl