# Comprehensive Reporting System

Advanced reporting system with templates, configurations, scheduling, and multi-format generation capabilities for the RAS Dashboard API.

## 🎯 Overview

The Reporting System provides: - **Template Management** - Reusable report templates with configurable parameters - **Configuration Management** - Saved report configurations for consistent generation - **Schedule Management** - Automated report generation with flexible scheduling - **Multi-Format Generation** - Support for PDF, Excel, CSV, JSON, HTML, Word, and PowerPoint - **File Management** - Secure file storage and download capabilities - **Analytics & Tracking** - Comprehensive usage analytics and performance monitoring - **Sharing & Collaboration** - Report sharing with access controls and subscriptions

## 🏗️ Database Schema

### Core Tables

-- Report Templates: Reusable report templates  
report\_templates (id, name, description, module, template\_data, is\_system,  
 created\_by, created\_at, updated\_at)  
  
-- Report Configurations: Saved report configurations  
report\_configurations (id, name, template\_id, parameters, created\_by,  
 created\_at, updated\_at)  
  
-- Report Schedules: Automated report scheduling  
report\_schedules (id, name, configuration\_id, frequency, cron\_expression,  
 next\_run, last\_run, recipients, delivery\_method, active,  
 timezone, created\_by, created\_at, updated\_at)  
  
-- Reports: Generated reports  
reports (id, name, description, type, status, format, parameters, file\_path,  
 file\_size, generated\_at, generated\_by, scheduled\_for, is\_recurring,  
 recurring\_schedule, last\_run\_at, next\_run\_at, error\_message,  
 template\_id, configuration\_id, schedule\_id, download\_count,  
 expires\_at, metadata, created\_at, updated\_at)  
  
-- Report Executions: Execution history and performance tracking  
report\_executions (id, report\_id, schedule\_id, status, started\_at,  
 completed\_at, duration, record\_count, file\_size,  
 error\_message, executed\_by, metadata, created\_at)  
  
-- Report Shares: Report sharing and access control  
report\_shares (id, report\_id, shared\_with, shared\_by, access\_level,  
 expires\_at, access\_count, last\_accessed\_at, is\_active,  
 share\_token, metadata, created\_at, updated\_at)  
  
-- Report Subscriptions: User subscriptions to reports  
report\_subscriptions (id, user\_id, report\_id, schedule\_id, delivery\_method,  
 delivery\_address, is\_active, last\_delivered\_at,  
 delivery\_count, preferences, created\_at, updated\_at)  
  
-- Report Analytics: Usage analytics and performance metrics  
report\_analytics (id, report\_id, template\_id, date, generation\_count,  
 download\_count, share\_count, view\_count,  
 average\_generation\_time, total\_file\_size, error\_count,  
 unique\_users, metadata, created\_at, updated\_at)

### Relationships

Users ←→ ReportTemplates (created\_by)  
Users ←→ ReportConfigurations (created\_by)  
Users ←→ ReportSchedules (created\_by)  
Users ←→ Reports (generated\_by)  
Users ←→ ReportExecutions (executed\_by)  
Users ←→ ReportShares (shared\_with, shared\_by)  
Users ←→ ReportSubscriptions (user\_id)  
  
ReportTemplates ←→ ReportConfigurations (template\_id)  
ReportConfigurations ←→ ReportSchedules (configuration\_id)  
ReportTemplates ←→ Reports (template\_id)  
ReportConfigurations ←→ Reports (configuration\_id)  
ReportSchedules ←→ Reports (schedule\_id)  
Reports ←→ ReportExecutions (report\_id)  
ReportSchedules ←→ ReportExecutions (schedule\_id)  
Reports ←→ ReportShares (report\_id)  
Reports ←→ ReportSubscriptions (report\_id)  
ReportSchedules ←→ ReportSubscriptions (schedule\_id)

## 📊 Report Types

### Supported Report Types

const REPORT\_TYPES = [  
 'dashboard', // Dashboard summaries and KPIs  
 'metrics', // System and application metrics  
 'analytics', // Data analytics and insights  
 'compliance', // Compliance and regulatory reports  
 'audit', // Audit trails and logs  
 'security', // Security incidents and assessments  
 'asset', // Asset inventory and management  
 'vulnerability', // Vulnerability assessments and scans  
 'policy', // Policy compliance and coverage  
 'procedure', // Procedure execution and compliance  
 'user\_activity', // User activity and behavior  
 'system\_performance', // System performance metrics  
 'financial', // Financial and cost reports  
 'operational', // Operational metrics and KPIs  
 'custom' // Custom report types  
];

### Report Status Workflow

const REPORT\_STATUSES = [  
 'draft', // Initial state, not yet generated  
 'generating', // Currently being generated  
 'completed', // Successfully generated  
 'failed', // Generation failed  
 'scheduled', // Scheduled for future generation  
 'cancelled', // Generation cancelled  
 'expired' // Report has expired  
];

### Supported Output Formats

const REPORT\_FORMATS = [  
 'pdf', // Portable Document Format  
 'excel', // Microsoft Excel (.xlsx)  
 'csv', // Comma-Separated Values  
 'json', // JavaScript Object Notation  
 'html', // HyperText Markup Language  
 'word', // Microsoft Word (.docx)  
 'powerpoint' // Microsoft PowerPoint (.pptx)  
];

## 📝 Template Management

### Template Structure

const reportTemplate = {  
 name: 'Monthly Security Report Template',  
 description: 'Comprehensive monthly security assessment template',  
 module: 'security',  
 templateData: {  
 sections: [  
 'executive\_summary',  
 'threat\_landscape',  
 'incident\_analysis',  
 'vulnerability\_assessment',  
 'compliance\_status',  
 'recommendations'  
 ],  
 parameters: {  
 dateRange: { type: 'string', default: '30d', required: true },  
 includeTrends: { type: 'boolean', default: true },  
 severityFilter: { type: 'array', default: ['high', 'critical'] },  
 outputFormat: { type: 'string', default: 'pdf' }  
 },  
 queries: {  
 incidents: 'SELECT \* FROM security\_incidents WHERE created\_at >= ?',  
 vulnerabilities: 'SELECT \* FROM vulnerabilities WHERE severity IN (?)',  
 compliance: 'SELECT \* FROM compliance\_checks WHERE status = ?'  
 },  
 formatting: {  
 title: 'Monthly Security Report - {{month}} {{year}}',  
 logo: '/assets/company-logo.png',  
 footer: 'Confidential - Internal Use Only',  
 colors: {  
 primary: '#1f2937',  
 secondary: '#3b82f6',  
 success: '#10b981',  
 warning: '#f59e0b',  
 danger: '#ef4444'  
 }  
 }  
 },  
 isSystem: false  
};

### Template Categories

const templateCategories = {  
 'security': {  
 description: 'Security-related reports and assessments',  
 commonParameters: ['dateRange', 'severityFilter', 'assetScope'],  
 defaultFormat: 'pdf'  
 },  
 'compliance': {  
 description: 'Regulatory compliance and audit reports',  
 commonParameters: ['framework', 'scope', 'assessmentDate'],  
 defaultFormat: 'excel'  
 },  
 'operational': {  
 description: 'Operational metrics and performance reports',  
 commonParameters: ['timeframe', 'departments', 'kpiSelection'],  
 defaultFormat: 'html'  
 },  
 'financial': {  
 description: 'Financial analysis and cost reports',  
 commonParameters: ['fiscalPeriod', 'costCenters', 'currency'],  
 defaultFormat: 'excel'  
 }  
};

## ⚙️ Configuration Management

### Configuration Structure

const reportConfiguration = {  
 name: 'Weekly Security Dashboard',  
 templateId: 1,  
 parameters: {  
 dateRange: '7d',  
 includeTrends: true,  
 severityFilter: ['high', 'critical'],  
 outputFormat: 'pdf',  
 recipients: ['security-team@company.com'],  
 customFilters: {  
 assetTypes: ['server', 'workstation'],  
 locations: ['datacenter-1', 'office-main'],  
 businessUnits: ['IT', 'Finance']  
 },  
 formatting: {  
 includeCharts: true,  
 chartTypes: ['bar', 'line', 'pie'],  
 pageOrientation: 'portrait',  
 fontSize: 12,  
 margins: { top: 20, bottom: 20, left: 15, right: 15 }  
 }  
 }  
};

### Parameter Validation

const parameterValidation = {  
 dateRange: {  
 type: 'string',  
 pattern: /^\d+[dwmy]$/, // e.g., 7d, 4w, 1m, 1y  
 required: true,  
 description: 'Time range for data collection'  
 },  
 severityFilter: {  
 type: 'array',  
 items: { enum: ['low', 'medium', 'high', 'critical'] },  
 minItems: 1,  
 description: 'Severity levels to include'  
 },  
 outputFormat: {  
 type: 'string',  
 enum: ['pdf', 'excel', 'csv', 'json', 'html'],  
 default: 'pdf',  
 description: 'Output format for the report'  
 },  
 includeCharts: {  
 type: 'boolean',  
 default: true,  
 description: 'Whether to include charts and visualizations'  
 }  
};

## 📅 Schedule Management

### Schedule Frequencies

const SCHEDULE\_FREQUENCIES = [  
 'once', // One-time execution  
 'daily', // Every day  
 'weekly', // Every week  
 'monthly', // Every month  
 'quarterly', // Every quarter  
 'yearly', // Every year  
 'custom' // Custom cron expression  
];

### Schedule Configuration

const reportSchedule = {  
 name: 'Daily Security Summary',  
 configurationId: 1,  
 frequency: 'daily',  
 cronExpression: '0 8 \* \* \*', // 8 AM daily  
 nextRun: '2024-01-16T08:00:00Z',  
 lastRun: '2024-01-15T08:00:00Z',  
 recipients: [  
 'security-team@company.com',  
 'ciso@company.com',  
 'operations@company.com'  
 ],  
 deliveryMethod: 'email',  
 active: true,  
 timezone: 'America/New\_York'  
};

### Cron Expression Examples

const cronExamples = {  
 'daily\_8am': '0 8 \* \* \*', // Every day at 8 AM  
 'weekly\_monday': '0 9 \* \* 1', // Every Monday at 9 AM  
 'monthly\_first': '0 10 1 \* \*', // First day of month at 10 AM  
 'quarterly': '0 9 1 1,4,7,10 \*', // Quarterly on 1st at 9 AM  
 'business\_hours': '0 9-17 \* \* 1-5', // Every hour 9-5, Mon-Fri  
 'end\_of\_month': '0 18 L \* \*' // Last day of month at 6 PM  
};

## 🔄 Report Generation Process

### Generation Workflow

1. Request Validation →  
2. Template/Configuration Loading →  
3. Parameter Processing →  
4. Data Collection →  
5. Content Generation →  
6. Format Conversion →  
7. File Storage →  
8. Notification Delivery →  
9. Analytics Recording

### Data Collection Methods

const dataCollectionMethods = {  
 database: {  
 description: 'Direct database queries',  
 example: 'SELECT \* FROM vulnerabilities WHERE severity = ?'  
 },  
 api: {  
 description: 'External API calls',  
 example: 'GET /api/v1/metrics?timeframe=30d'  
 },  
 file: {  
 description: 'File system data',  
 example: 'Read log files from /var/log/security/'  
 },  
 calculation: {  
 description: 'Computed metrics',  
 example: 'Calculate risk scores from multiple data sources'  
 }  
};

### Content Generation

const contentGeneration = {  
 sections: {  
 header: 'Company logo, report title, generation date',  
 summary: 'Executive summary with key findings',  
 metrics: 'Key performance indicators and statistics',  
 charts: 'Visual representations of data',  
 details: 'Detailed findings and analysis',  
 recommendations: 'Actionable recommendations',  
 appendix: 'Supporting data and references',  
 footer: 'Confidentiality notice and page numbers'  
 },  
 formatting: {  
 fonts: ['Arial', 'Helvetica', 'Times New Roman'],  
 colors: 'Corporate color scheme',  
 layout: 'Professional business format',  
 branding: 'Company logo and styling'  
 }  
};

## 🚀 API Endpoints

### Report Template Management (5 endpoints)

// Template CRUD operations  
POST /api/v1/reports/templates // Create template  
GET /api/v1/reports/templates // Get all templates with filtering  
GET /api/v1/reports/templates/:id // Get template by ID  
PUT /api/v1/reports/templates/:id // Update template  
DELETE /api/v1/reports/templates/:id // Delete template

### Report Configuration Management (3 endpoints)

// Configuration CRUD operations  
POST /api/v1/reports/configurations // Create configuration  
GET /api/v1/reports/configurations // Get all configurations with filtering  
GET /api/v1/reports/configurations/:id // Get configuration by ID

### Report Generation & Management (5 endpoints)

// Report operations  
POST /api/v1/reports // Generate report  
GET /api/v1/reports // Get all reports with filtering  
GET /api/v1/reports/:id // Get report by ID  
GET /api/v1/reports/:id/download // Download report file  
DELETE /api/v1/reports/:id // Delete report

## 🛠️ Usage Examples

### Creating a Report Template

const templateData = {  
 name: 'Vulnerability Assessment Report',  
 description: 'Comprehensive vulnerability assessment with risk analysis',  
 module: 'security',  
 templateData: {  
 sections: [  
 'executive\_summary',  
 'methodology',  
 'findings',  
 'risk\_analysis',  
 'remediation\_plan'  
 ],  
 parameters: {  
 scanType: { type: 'string', enum: ['full', 'targeted'], default: 'full' },  
 severityThreshold: { type: 'string', enum: ['low', 'medium', 'high'], default: 'medium' },  
 includeRemediation: { type: 'boolean', default: true },  
 assetScope: { type: 'array', items: { type: 'string' } }  
 },  
 queries: {  
 vulnerabilities: `  
 SELECT v.\*, a.name as asset\_name, a.type as asset\_type  
 FROM vulnerabilities v  
 JOIN assets a ON v.asset\_id = a.id  
 WHERE v.severity >= ? AND v.status = 'open'  
 ORDER BY v.cvss\_score DESC  
 `,  
 riskScores: `  
 SELECT asset\_id, AVG(cvss\_score) as avg\_risk  
 FROM vulnerabilities  
 WHERE status = 'open'  
 GROUP BY asset\_id  
 ORDER BY avg\_risk DESC  
 `  
 },  
 formatting: {  
 title: 'Vulnerability Assessment Report - {{date}}',  
 classification: 'CONFIDENTIAL',  
 pageNumbers: true,  
 tableOfContents: true  
 }  
 },  
 isSystem: false  
};  
  
const template = await reportService.createTemplate(templateData, userId);

### Creating a Report Configuration

const configurationData = {  
 name: 'Monthly Vulnerability Report - Critical Assets',  
 templateId: template.id,  
 parameters: {  
 scanType: 'targeted',  
 severityThreshold: 'high',  
 includeRemediation: true,  
 assetScope: ['production-servers', 'database-servers', 'web-applications'],  
 customFilters: {  
 businessCritical: true,  
 publicFacing: true,  
 dataClassification: ['confidential', 'restricted']  
 },  
 outputSettings: {  
 format: 'pdf',  
 includeCharts: true,  
 detailLevel: 'comprehensive',  
 appendices: ['raw\_data', 'methodology', 'references']  
 }  
 }  
};  
  
const configuration = await reportService.createConfiguration(configurationData, userId);

### Generating a Report

const reportData = {  
 name: 'Q1 2024 Security Assessment',  
 description: 'Quarterly security assessment covering all critical systems',  
 type: 'security',  
 format: 'pdf',  
 parameters: {  
 quarter: 'Q1',  
 year: 2024,  
 scope: 'all-systems',  
 includeExecutiveSummary: true,  
 includeRecommendations: true,  
 detailLevel: 'comprehensive'  
 },  
 templateId: template.id,  
 configurationId: configuration.id,  
 expiresAt: new Date(Date.now() + 90 \* 24 \* 60 \* 60 \* 1000), // 90 days  
 metadata: {  
 requestedBy: 'CISO',  
 priority: 'high',  
 distribution: 'executive-team',  
 classification: 'confidential'  
 }  
};  
  
const report = await reportService.generateReport(reportData, userId);  
console.log(`Report generation started: ${report.id}`);

### Advanced Filtering and Search

// Get reports with complex filtering  
const reports = await reportService.getAllReports({  
 type: 'security',  
 status: 'completed',  
 generatedBy: userId,  
 startDate: '2024-01-01T00:00:00Z',  
 endDate: '2024-03-31T23:59:59Z',  
 search: 'vulnerability assessment'  
}, {  
 page: 1,  
 limit: 20,  
 sortBy: 'generatedAt',  
 sortOrder: 'desc'  
});  
  
console.log(`Found ${reports.pagination.totalCount} reports`);  
reports.data.forEach(report => {  
 console.log(`- ${report.name} (${report.type}) - ${report.status}`);  
});

## 📊 Analytics and Monitoring

### Report Analytics

const reportAnalytics = {  
 overall: {  
 totalReports: 1250,  
 completedReports: 1180,  
 failedReports: 45,  
 scheduledReports: 25,  
 successRate: 94.4,  
 averageGenerationTime: 45000, // milliseconds  
 totalFileSize: 2500000000, // bytes  
 totalDownloads: 3200  
 },  
 byType: [  
 { type: 'security', count: 450, avgSize: 2500000, avgTime: 60000 },  
 { type: 'compliance', count: 320, avgSize: 1800000, avgTime: 40000 },  
 { type: 'operational', count: 280, avgSize: 1200000, avgTime: 30000 }  
 ],  
 byFormat: [  
 { format: 'pdf', count: 650, percentage: 52.0 },  
 { format: 'excel', count: 380, percentage: 30.4 },  
 { format: 'csv', count: 150, percentage: 12.0 },  
 { format: 'html', count: 70, percentage: 5.6 }  
 ],  
 performance: {  
 averageGenerationTime: 45000,  
 p95GenerationTime: 120000,  
 p99GenerationTime: 300000,  
 errorRate: 3.6,  
 peakHours: ['09:00', '14:00', '16:00']  
 }  
};

This comprehensive reporting system provides enterprise-grade report generation capabilities with advanced template management, flexible scheduling, multi-format output, and robust security controls for modern organizations.