Contributing Guide (CONTRIBUTING.md)

Thank you for considering contributing to the Coverity Connect MCP Server! We welcome contributions from the community.

🤝 How to Contribute

Reporting Issues

- Use GitHub Issues to report bugs or request features
- Search existing issues before creating new ones
- Provide detailed reproduction steps for bugs
- Include environment information (OS, Python version, etc.)

Submitting Changes

1. **Fork** the repository
2. **Create a branch** for your feature (`git checkout -b feature/amazing-feature`)
3. **Make your changes** following our coding standards
4. **Add tests** for your changes
5. **Run the test suite** to ensure everything works
6. **Commit** with clear messages (`git commit -m 'Add amazing feature'`)
7. **Push** to your branch (`git push origin feature/amazing-feature`)
8. **Submit a Pull Request**

Development Setup

```bash
git clone https://github.com/keides2/coverity-connect-mcp.git
cd coverity-connect-mcp
python -m venv venv
source venv/bin/activate  # Windows: venv\Scripts\activate
pip install -e ".[dev]"
pre-commit install
```

Running Tests

```bash
```

```
# Unit tests
pytest tests/

# With coverage
pytest --cov=coverity_mcp_server tests/

# Integration tests
pytest tests/ -m integration
```

Code Standards

- Follow PEP 8 style guidelines
- Use Black for code formatting
- Use isort for import sorting
- Use mypy for type checking
- Write docstrings for all public functions
- Maintain test coverage above 80%

Security Policy (SECURITY.md)

🔒 Security Policy

Supported Versions

Version	Supported
1.x.x	✅
< 1.0	❌

Reporting Security Vulnerabilities

DO NOT report security vulnerabilities through public GitHub issues.

Instead, please send an email to: security@your-domain.com

Include the following information:

- Type of issue (e.g. buffer overflow, SQL injection, cross-site scripting, etc.)
- Full paths of source file(s) related to the manifestation of the issue
- The location of the affected source code (tag/branch/commit or direct URL)
- Any special configuration required to reproduce the issue
- Step-by-step instructions to reproduce the issue

- Proof-of-concept or exploit code (if possible)
- Impact of the issue, including how an attacker might exploit the issue

Security Considerations

This MCP server connects to Coverity Connect servers and handles:

- Authentication credentials (auth keys)
- Corporate network access
- Static analysis results
- Potentially sensitive source code information

Best Practices:

- Always use environment variables for credentials
- Use HTTPS connections to Coverity Connect
- Keep dependencies updated
- Follow principle of least privilege
- Regularly rotate authentication keys

Changelog (CHANGELOG.md)

Changelog

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

[Unreleased]

Added

- Initial MCP server implementation
- Coverity Connect SOAP API integration
- Project and stream management tools
- Snapshot analysis capabilities
- Security vulnerability analysis
- Quality reporting features
- Docker containerization
- GitHub Actions CI/CD

Security

- Secure authentication key handling
- Environment variable configuration
- Input validation and sanitization

[1.0.0] - 2025-01-XX

Added

- First stable release
- Complete MCP protocol implementation
- Full Coverity Connect integration
- Comprehensive documentation
- Testing suite with 90%+ coverage
- Multi-platform support (Windows/macOS/Linux)

Issue Templates

Bug Report (.github/ISSUE_TEMPLATE/bug_report.md)

---

name: Bug report about: Create a report to help us improve title: '[BUG] ' labels: 'bug' assignees: ''

Describe the bug A clear and concise description of what the bug is.

To Reproduce Steps to reproduce the behavior:

1. Configure with '...'
2. Run command '...'
3. See error

Expected behavior A clear and concise description of what you expected to happen.

Environment:

- OS: [e.g. Windows 11, macOS 14, Ubuntu 22.04]
- Python version: [e.g. 3.11.5]
- Package version: [e.g. 1.0.0]
- Coverity Connect version: [e.g. 2023.6.0]

Configuration:

```
yaml

# Your configuration (remove sensitive data)
```

Logs:

```
# Relevant log output
```

Additional context Add any other context about the problem here.

Feature Request (.github/ISSUE_TEMPLATE/feature_request.md)

---

name: Feature request about: Suggest an idea for this project title: '[FEATURE] ' labels: 'enhancement' assignees: ''

Is your feature request related to a problem? Please describe. A clear and concise description of what the problem is.

Describe the solution you'd like A clear and concise description of what you want to happen.

Describe alternatives you've considered A clear and concise description of any alternative solutions or features you've considered.

Use case Describe how this feature would be used and who would benefit from it.

Additional context Add any other context or screenshots about the feature request here.

Security Issue (.github/ISSUE_TEMPLATE/security_issue.md)

---

name: Security Issue about: Report a security vulnerability (use email for sensitive issues) title: '[SECURITY] ' labels: 'security' assignees: ''

⚠️ SECURITY NOTICE ⚠️

For sensitive security issues, please DO NOT use this public issue tracker. Instead, email: security@your-domain.com

For non-sensitive security improvements:

Security concern Describe the security issue or improvement suggestion.

Impact What could happen if this issue is exploited?

Suggested fix If you have ideas for how to fix this, please share them.

Environment

- Deployment type: [e.g. Docker, local, cloud]
- Network configuration: [e.g. corporate proxy, direct internet]

Pull Request Template (.github/pull_request_template.md)

📋 Pull Request

Description Brief description of changes made.

Type of Change

□ Bug fix (non-breaking change which fixes an issue)
□ New feature (non-breaking change which adds functionality)
□ Breaking change (fix or feature that would cause existing functionality to not work as expected)
□ Documentation update
□ Performance improvement
□ Code refactoring

Testing

□ Unit tests pass
□ Integration tests pass
□ Manual testing completed
□ Added new tests for changes

Checklist

□ Code follows project style guidelines
□ Self-review of code completed
□ Code is commented, particularly in hard-to-understand areas
□ Documentation updated
□ No new warnings introduced
□ All CI checks pass

Screenshots/Output If applicable, add screenshots or command output.

Related Issues Fixes #(issue number)

Additional Notes Any additional information or context.