

Coverity Connect MCP Server 機能仕様書

Version 1.0.0

作成日: 2025年7月21日

更新日: 2025年7月21日

📋 概要

Coverity Connect MCP Serverは、Black Duck Coverity Connect静的解析プラットフォームとAIアシスタント（Claude Desktop等）間のシームレスな統合を提供するModel Context Protocol (MCP) サーバーです。

目的

- Coverity Connectへの自然言語アクセスの実現
- 静的解析結果の効率的な分析・レポート生成
- セキュリティ脆弱性の迅速な検出・対応
- CI/CDパイプラインとの統合支援

アーキテクチャ

Claude Desktop/AI Assistant
↓ (MCP Protocol)
Coverity Connect MCP Server
↓ (REST/SOAP API)
Black Duck Coverity Connect

🔧 実装技術スタック

主要技術

- Python 3.8+**: 実行環境
- FastMCP**: MCPサーバーフレームワーク
- aiohttp**: 非同期HTTPクライアント
- suds-community**: SOAP APIクライアント
- Click**: CLI インターフェース

依存関係

python

```
dependencies = [  
    "mcp>=1.0.0",  
    "fastmcp>=0.1.0",  
    "suds-community>=1.1.2",  
    "requests>=2.31.0",  
    "pandas>=1.5.0",  
    "pydantic>=2.0.0",  
    "aiofiles>=23.0.0",  
    "python-dotenv>=1.0.0",  
    "click>=8.0.0",  
    "rich>=13.0.0",  
    "PyYAML>=6.0"  
]
```

🔧 システム機能一覧

1. プロジェクト管理機能

1.1 プロジェクト一覧取得 (`list_projects`)

機能説明: Coverity Connect内の全プロジェクトを取得

入力パラメータ: なし

出力形式:

```
json  
  
[  
  {  
    "projectKey": "PROJ001",  
    "projectName": "WebApplication",  
    "description": "メインWebアプリケーションプロジェクト",  
    "createdDate": "2024-01-15T10:30:00Z",  
    "lastModified": "2024-07-20T15:45:00Z",  
    "streams": ["main", "develop", "release"]  
  }  
]
```

使用例:

Coverityの全プロジェクト一覧を表示してください

1.2 プロジェクト詳細取得 (`get_project_summary`)

機能説明: 指定プロジェクトの包括的な情報と統計を取得

入力パラメータ:

- `project_id` (必須): プロジェクト識別子

出力形式:

```
json
{
  "project": {
    "projectKey": "PROJ001",
    "projectName": "WebApplication",
    "description": "メインWebアプリケーションプロジェクト"
  },
  "streams": [
    {
      "stream_name": "main-stream",
      "total_defects": 15,
      "severity_breakdown": {
        "High": 3,
        "Medium": 8,
        "Low": 4
      },
      "status_breakdown": {
        "New": 10,
        "Triaged": 3,
        "Fixed": 2
      }
    }
  ],
  "total_streams": 3
}
```

2. ストリーム管理機能

2.1 ストリーム一覧取得 (`list_streams`)

機能説明: プロジェクト内のストリーム一覧を取得

入力パラメータ:

- `project_id` (任意): プロジェクトIDによるフィルタリング

出力形式:

```
json
```

```
[
  {
    "name": "main-stream",
    "description": "メイン開発ストリーム",
    "projectId": "WebApplication",
    "language": "MIXED"
  }
]
```

3. 欠陥検索・分析機能

3.1 欠陥検索 (`search_defects`)

機能説明: 高度なフィルタリング機能を持つ欠陥検索

入力パラメータ:

- `query` (任意): 汎用検索クエリ
- `stream_id` (任意): ストリームIDフィルタ
- `checker` (任意): チェッカー名フィルタ (例: NULL_RETURNS)
- `severity` (任意): 重要度フィルタ (High, Medium, Low)
- `status` (任意): ステータスフィルタ (New, Triaged, Fixed等)
- `limit` (任意): 最大結果数 (デフォルト: 50)

出力形式:

```
json
[
  {
    "cid": "12345",
    "checkerName": "NULL_RETURNS",
    "displayType": "Null pointer dereference",
    "displayImpact": "High",
    "displayStatus": "New",
    "displayFile": "src/main.c",
    "displayFunction": "main",
    "firstDetected": "2024-01-15T10:00:00Z",
    "streamId": "main-stream"
  }
]
```

使用例:

main-streamでHigh重要度のセキュリティ欠陥を検索してください

3.2 欠陥詳細取得 (`get_defect_details`)

機能説明: CID (Coverity Issue Identifier) による詳細欠陥情報の取得

入力パラメータ:

- `cid` (必須): Coverity Issue Identifier

出力形式:

```
json
{
  "cid": "12345",
  "checkerName": "NULL_RETURNS",
  "displayType": "Null pointer dereference",
  "displayImpact": "High",
  "displayStatus": "New",
  "displayFile": "src/main.c",
  "displayFunction": "main",
  "firstDetected": "2024-01-15T10:00:00Z",
  "streamId": "main-stream",
  "occurrenceCount": 1,
  "events": [
    {
      "eventNumber": 1,
      "eventTag": "assignment",
      "eventDescription": "Null assignment detected",
      "fileName": "src/main.c",
      "lineNumber": 42
    }
  ]
}
```

4. ユーザー管理機能 ✨ 新機能

4.1 ユーザー一覧取得 (`list_users`)

機能説明: Coverity Connect内の全ユーザー情報を取得

入力パラメータ:

- `include_disabled` (任意): 無効化ユーザーを含める (デフォルト: False)
- `limit` (任意): 最大取得ユーザー数 (デフォルト: 200)

出力形式:

```
json
[
  {
    "name": "admin",
    "email": "admin@company.com",
    "familyName": "Administrator",
    "givenName": "System",
    "disabled": false,
    "locked": false,
    "superUser": true,
    "groupNames": ["Administrators", "Users"],
    "roleAssignments": [
      {
        "roleName": "administrator",
        "scope": "global",
        "username": "admin"
      }
    ],
    "lastLogin": "2024-07-21T10:00:00Z",
    "dateCreated": "2024-01-01T00:00:00Z",
    "local": true
  }
]
```

使用例:

Coverityの全ユーザー一覧を表示してください

4.2 ユーザー詳細取得 (get_user_details)

機能説明: 指定ユーザーの詳細情報を取得

入力パラメータ:

- username (必須): ユーザー名

出力形式:

```
json
```

```
{
  "name": "developer1",
  "email": "dev1@company.com",
  "familyName": "開発",
  "givenName": "太郎",
  "disabled": false,
  "locked": false,
  "superUser": false,
  "groupNames": ["Users"],
  "roleAssignments": [
    {
      "roleName": "developer",
      "scope": "global",
      "username": "developer1"
    }
  ],
  "lastLogin": "2024-07-20T15:30:00Z",
  "dateCreated": "2024-02-01T00:00:00Z",
  "local": true
}
```

4.3 ユーザー権限情報取得 (`get_user_roles`)

機能説明: 指定ユーザーのロールと権限情報を詳細に取得

入力パラメータ:

- `username` (必須): 権限を調べるユーザー名

出力形式:

json

```
{
  "username": "projectowner1",
  "superUser": false,
  "groups": ["Users"],
  "roles": [
    {
      "roleName": "projectOwner",
      "scope": "project",
      "roleAssignmentType": "user",
      "description": "プロジェクトの所有者権限"
    }
  ],
  "status": {
    "disabled": false,
    "locked": false,
    "local": true
  },
  "lastLogin": "2024-07-19T09:15:00Z",
  "dateCreated": "2024-03-01T00:00:00Z"
}
```

ロール説明:

- `administrator`: システム全体の管理権限
- `projectOwner`: プロジェクトの所有者権限
- `developer`: 開発者権限
- `analyst`: 分析者権限
- `viewer`: 閲覧権限

使用例:

ユーザー 'developer1' の権限とロール情報を教えてください

5. リソース機能

5.1 プロジェクト設定リソース

URI: `coverity://projects/{project_id}/config` **機能:** 指定プロジェクトの設定情報にアクセス

5.2 ストリーム欠陥リソース

URI: `coverity://streams/{stream_id}/defects` **機能:** 指定ストリームの欠陥情報にアクセス

環境変数設定

```
bash

# 必須設定
export COVERITY_HOST="your-coverity-server.com"
export COVERITY_PORT="8080"
export COVERITY_SSL="True"
export COVAUTHUSER="your-username"
export COVAUTHKEY="your-auth-key"

# オプション設定
export COVERITY_BASE_DIR="/path/to/workspace"
export PROXY_HOST="proxy-server.com"
export PROXY_PORT="3128"
export LOG_LEVEL="INFO"
```

SSL/TLS設定

- デフォルトでHTTPS接続を使用
- 自己署名証明書への対応
- 企業プロキシ環境への対応

権限モデル

- Coverity Connectの既存認証システムを使用
- ユーザーごとのアクセス制御
- ロールベースの権限管理

エラーハンドリング

エラーコード

コード	メッセージ	説明
401	Authentication failed	認証失敗
404	Resource not found	リソースが見つからない
500	Server error	サーバー内部エラー
503	Service unavailable	Coverity Connect接続不可

エラー応答形式

```
json
```

```
{  
  "error": "Error message describing the issue"  
}
```

テスト方式

開発環境テスト

```
bash  
  
# ダミーサーバー起動  
python examples/development/mock_server.py  
  
# MCPサーバー起動  
python -m coverity_mcp_server
```

単体テスト

```
bash  
  
# テスト実行  
pytest tests/  
  
# カバレッジテスト  
pytest --cov=coverity_mcp_server tests/
```

統合テスト

```
bash  
  
# 実際のCoverity Connectとの接続テスト  
pytest tests/ -m integration
```

CI/CDパイプライン統合

GitHub Actions

- 自動テスト実行
- コード品質チェック
- パッケージビルド・公開

Docker対応

```
dockerfile
```

```
FROM python:3.11-slim
COPY ./app
WORKDIR /app
RUN pip install -e .
CMD ["coverity-mcp-server"]
```

パフォーマンス仕様

レスポンス時間

- プロジェクト一覧取得: < 5秒
- 欠陥検索（100件）: < 10秒
- ユーザー一覧取得: < 3秒
- 詳細情報取得: < 2秒

スループット

- 同時接続数: 10接続
- 最大クエリ数: 1000/時間
- メモリ使用量: < 512MB

制限事項

- 欠陥検索結果の上限: 1000件
- ユーザー一覧の上限: 200件
- タイムアウト設定: 30秒

使用シナリオ例

シナリオ 1: セキュリティ脆弱性の分析

1. "WebApplicationプロジェクトのHigh重要度セキュリティ欠陥を分析してください"
2. システムが自動的に:
 - プロジェクト検索
 - 重要度フィルタリング
 - セキュリティ関連欠陥抽出
 - 対策提案

シナリオ 2: プロジェクト品質レポート生成

1. "MyProjectの品質サマリーを生成してください"
2. システムが提供:
 - 全体統計
 - ストリーム別分析
 - 優先対応項目
 - トレンド分析

シナリオ 3: ユーザー権限管理

1. "開発チームのメンバーと権限を確認してください"
2. システムが表示:
 - ユーザー一覧
 - ロール・権限詳細
 - アクセス状況
 - 権限不備の警告

シナリオ 4: CI/CD統合

1. "main-streamの最新ビルド結果を分析してください"
2. システムが実行:
 - 最新スナップショット取得
 - 新規欠陥識別
 - 修正済み欠陥確認
 - ビルド品質判定

今後の拡張計画

Version 1.1

- ☐ 高度な欠陥フィルタリング
- ☐ カスタムビュー機能
- ☐ 欠陥トレンド分析

Version 1.2

- ☐ マルチテナント対応
- ☐ 拡張ユーザー管理
- ☐ チーム・グループ管理

Version 1.3

- ☐ REST API対応強化
- ☐ GraphQL API対応
- ☐ リアルタイム通知

Version 2.0

- ☐ ML欠陥優先度付け
- ☐ プラグインアーキテクチャ
- ☐ サードパーティ統合

サポート・連絡先

技術サポート

- **GitHub Issues:** [Issues](#)
- **Discussions:** [Discussions](#)

ドキュメント

- **API Reference:** [docs/api.md](#)
- **Configuration Guide:** [docs/configuration.md](#)
- **Installation Guide:** [docs/installation.md](#)
- **Setup Guide:** [SETUP_GUIDE.md](#)

この機能仕様書は、Coverity Connect MCP Server v1.0.0の全機能を包括的に説明しています。最新のユーザー管理機能の追加により、セキュリティ監査、アクセス制御、チーム管理がより効率的になりました。

更新履歴:

- 2025-07-21: ユーザー管理機能 (list_users, get_user_details, get_user_roles) 追加
- 2025-07-21: 初版作成 (v1.0.0ベース機能仕様)