目录结构

```
dataset-xxx/
├── <project_name>_<version>/          # 每个漏洞项目目录（共 416 个）
│   ├── project_meta.json              # 项目元数据（名称、语言、框架、CVE 统计等）
│   ├── sast_standardized.json         # SAST 标准化扫描结果
│   ├── Dockerfile                     # 项目容器化构建文件
│   ├── docker-compose.yaml/yml        # 服务编排配置
│   ├── deployment.json                # 部署参数（端口、环境变量、启动超时等）
│   ├── task-deps/                     # 任务依赖资源（补丁、初始化脚本等）
│   └── findings/
│       └── <CVE-ID>/                  # 每个 CVE 一个子目录
│           └── verify_requirements/   # 验证与修复数据包
│               ├── one_issue.txt      # 漏洞摘要（危害等级、CWE、修复版本等）
│               ├── phase1_decision.json    # Phase 1：漏洞确认决策
│               ├── phase2_verification.json # Phase 2：动态验证结果
│               ├── phase3_remediation.json  # Phase 3：修复方案评估
│               ├── result.json        # 整体评测结论
│               ├── fix.md             # 修复说明
│               ├── sast_standardized.json  # CVE 级 SAST 数据
│               ├── tests/             # PoC 测试用例（Python 脚本）
│               ├── logs/              # 执行日志
│               └── time.txt           # 耗时记录
```

数据字段说明

`project_meta.json`

字段	说明
`project_name`	项目名称
`version`	漏洞所在版本
`language`	主要编程语言
`framework`	使用框架（如 gin、rails、express 等）
`source_url`	上游仓库地址
`commit_id`	对应提交 hash
`total_findings`	该目录下 CVE 总数
`tp_count`	真阳性数量
`fp_count`	假阳性数量

`sast_standardized.json`（项目级）

字段	说明
`findings[].finding_id`	CVE ID
`findings[].cwe`	CWE 分类
`findings[].vuln_type`	漏洞类型（如 OS Command Injection）
`findings[].severity`	危害等级（HIGH / MEDIUM / LOW / CRITICAL）
`findings[].cvss`	CVSS 评分
`findings[].vul_pos`	漏洞位置列表（文件、行号、角色、代码片段）
`findings[].description`	漏洞描述（含 NVD 摘要与仓库交叉核对）
`findings[].recommendation`	修复建议