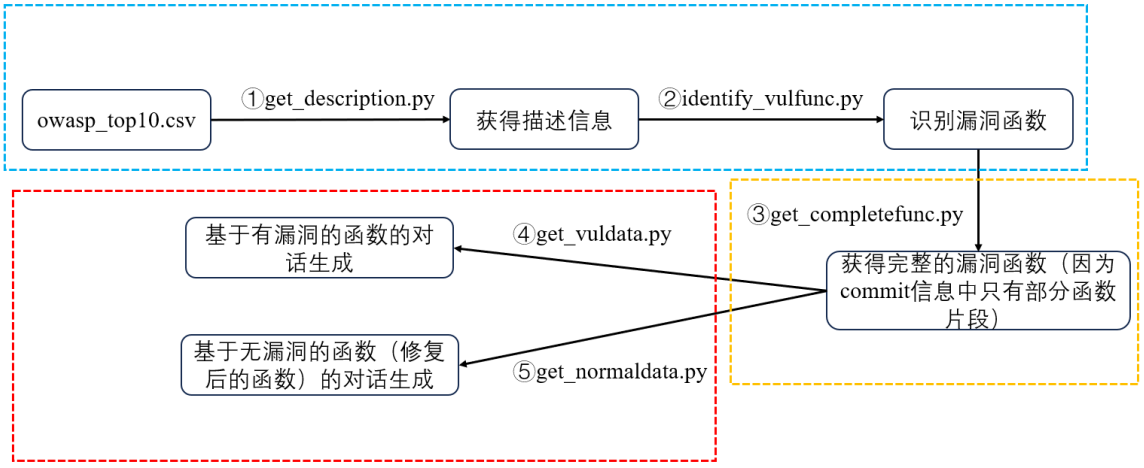
readme

- 目标：**自动化**地构造有监督的微调的数据集，预期输出示例如下：


```
{
    "instruction": "Can you briefly describe the purpose of this function?",
    "output": "The function ntlm_decode_u16l_str_hdr() decodes a 16-bit little-endian Unicode string header from an NTLM context int
    "input": "static int ntlm_decode_u16l_str_hdr(struct ntlm_ctx *ctx,\n\n                                    struct wire_field_hdr
},
{
    "instruction": "What's the most critical part of this function?",
    "output": "The most critical part of the function is where it checks the bounds of the string's offset and length against the bu
    "input": "static int ntlm_decode_u16l_str_hdr(struct ntlm_ctx *ctx,\n\n                                    struct wire_field_hdr
},
{
    "instruction": "Can you analyze the risks associated with the key part of this function?",
    "output": "The key part of the function validates the string's offset and length to prevent out-of-bounds access, reducing the r
    "input": "static int ntlm_decode_u16l_str_hdr(struct ntlm_ctx *ctx,\n\n                                    struct wire_field_hdr
},
```


- 流程图：

目前的框架



- get_description.py
  - [CVEProject/cvelistV5: CVE cache of the official CVE List in CVE JSON 5 format (github.com)]

    记录CVE信息的仓库
  - input："owasp_top10.csv" 和 上述 github仓库 clone 到本地的路径

    output：json(字典列表)


```
[
{
"CVEid": "",
"CWEid": "",
```

```
"commit_urls": "",
"diff": "",
"desc" : ""
},
{
"CVEid": "",
"CWEid": "",
"commit_urls": "",
"diff": "",
"desc" : ""
},
]
```


- identify_vulfunc.py
  - 准备一个openai api key
  - input: 上一步输出的json文件
  - output格式如下（json，字典列表）：


```
[
    {
        "CVE": "CVE-2023-46728",
        "commit": "https://github.com/squid-
cache/squid/commit/6ea12e8fb590ac6959e9356a81aa3370576568c3",
        "description": "Squid is a caching proxy for the Web supporting
HTTP, HTTPS, FTP, and more. Due to a NULL pointer dereference bug Squid is
vulnerable to a Denial of Service attack against Squid's Gopher gateway.
The gopher protocol is always available and enabled in Squid prior to
Squid 6.0.1. Responses triggering this bug are possible to be received
from any gopher server, even those without malicious intent. Gopher
support has been removed in Squid version 6.0.1. Users are advised to
upgrade. Users unable to upgrade should reject all gopher URL requests.",
        "vul_filename": "src/FwdState.cc",
        "vul_function": "FwdState::dispatch"
    },
    {
        "CVE": "CVE-2023-46728",
        "commit": "https://github.com/squid-
cache/squid/commit/6ea12e8fb590ac6959e9356a81aa3370576568c3",
        "description": "Squid is a caching proxy for the Web supporting
HTTP, HTTPS, FTP, and more. Due to a NULL pointer dereference bug Squid is
vulnerable to a Denial of Service attack against Squid's Gopher gateway.
The gopher protocol is always available and enabled in Squid prior to
Squid 6.0.1. Responses triggering this bug are possible to be received
from any gopher server, even those without malicious intent. Gopher
```

```
support has been removed in Squid version 6.0.1. Users are advised to
upgrade. Users unable to upgrade should reject all gopher URL requests.",
        "vul_filename": "src/FwdState.cc",
        "vul_function": "FwdState::dispatch"
    },
]
```

- get_completefunc.py：获得完成的漏洞函数，修复代码，以及diff （不依赖gpt，可在本地由一个人完成）

  输出为：json(字典列表)

```
[
{
"CVE":"",
"commit":"",
"description":"",
"vul_filename":"",
"vul_function":"",
"vul_code":"",
"fixed_code":"",
"diff":""
},
]
```

- get_vuldata.py

  input: get_completefunc.py输出的json(字典列表)

  output：最终对于有漏洞函数的分析对话

```
{
    "instruction": "Can you briefly describe the purpose of this function?",
    "output": "The function ntlm_decode_u16l_str_hdr() decodes a 16-bit little-endian Unicode string header from an NTLM context int
    "input": "static int ntlm_decode_u16l_str_hdr(struct ntlm_ctx *ctx,\n\n                                    struct wire_field_hdr
},
{
    "instruction": "What's the most critical part of this function?",
    "output": "The most critical part of the function is where it checks the bounds of the string's offset and length against the bu
    "input": "static int ntlm_decode_u16l_str_hdr(struct ntlm_ctx *ctx,\n\n                                    struct wire_field_hdr
},
{
    "instruction": "Can you analyze the risks associated with the key part of this function?",
    "output": "The key part of the function validates the string's offset and length to prevent out-of-bounds access, reducing the r
    "input": "static int ntlm_decode_u16l_str_hdr(struct ntlm_ctx *ctx,\n\n                                    struct wire_field_hdr
},
```

- get_normaldata.py

  input:get_completefunc.py输出的json(字典列表)

  output:最终对于无漏洞函数的分析对话

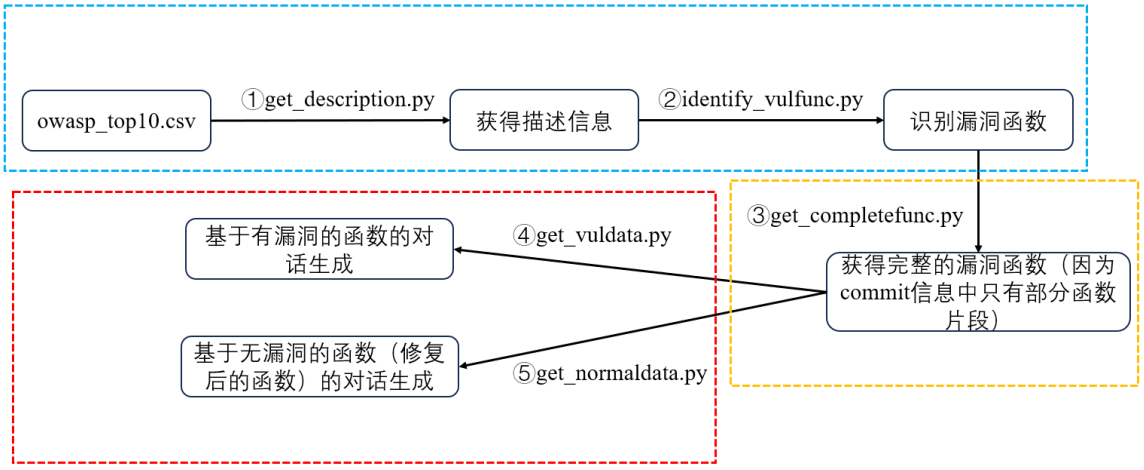
  格式同上

TODO:

- 设计合理的prompt
  - 获得审计报告的prompt
  - 基于审计报告获得对话的prompt
- 优化get_completefunc.py的分析
- 优化代码（包括但不限于规范代码，提高效率，重构等）
- ④get_vuldata.py 和⑤get_normaldata.py应该进行拆分，因为目前的实现是输入一个prompt获得对话内容；需要拆分成两轮对话（第一个prompt获得审计报告，第二个prompt获得对话）

之后批量处理时的分工：

目前的框架



- 蓝色框：多人gpt4并行处理，汇总结果给负责黄色框的同学分析
- 黄色框：一人本地分析（不涉及gpt调用，需要大量文件分析操作），将输出结果给负责红色框的同学分析
- 红色框：多人gpt4并行处理，得到可直接用于SFT训练的数据
- ps：多人并行需要的多个api，可以由一个账户下游分发多个api。