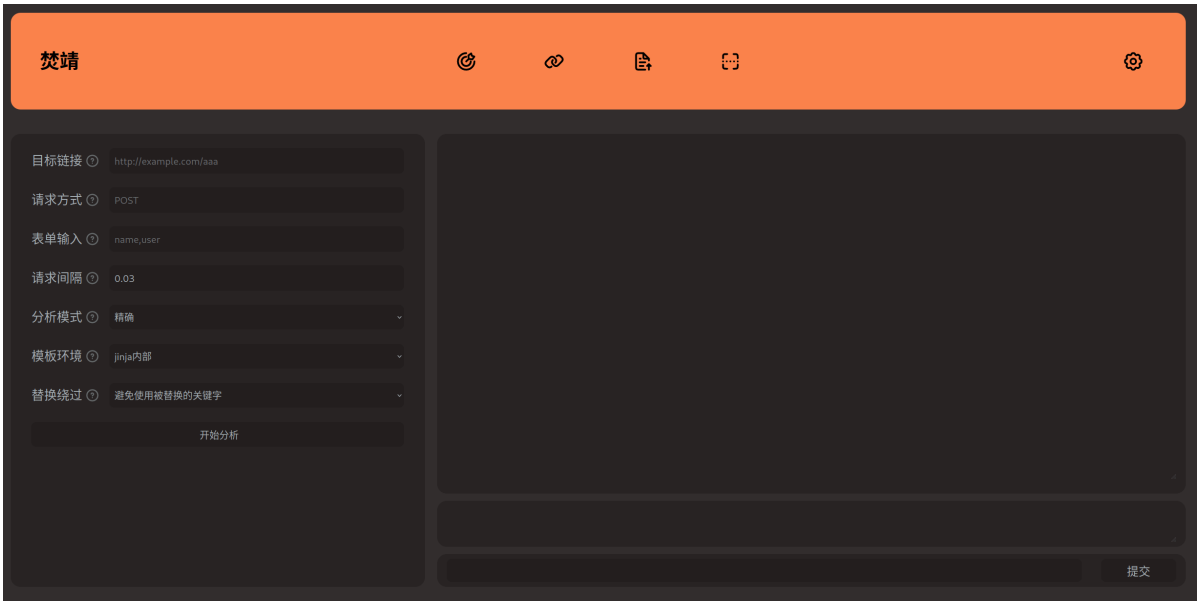
ssti的好工具 fenjing使用教程

webui

可以直接输入`python -m fenjing webui`启动webui，指定参数并自动攻击



在左边填入参数并点击开始分析，然后在右边输入命令即可

scan

在终端可以用scan功能，猜测某个页面的参数并自动攻击：

```
python -m fenjing scan --url 'http://xxxx:xxx/yyy'
```

crack

也可以用crack功能，手动指定参数进行攻击：

```
python -m fenjing crack --url 'http://xxxx:xxx/yyy' --detect-mode fast --inputs
aaa,bbb --method GET
```

这里提供了aaa和bbb两个参数进行攻击，并使用`--detect-mode fast`加速攻击速度

crack-request

还可以将HTTP请求写进一个文本文件里（比如说`req.txt`）然后进行攻击

文本文件内容如下：

```
GET /?name=PAYLOAD HTTP/1.1
Host: 127.0.0.1:5000
Connection: close
```

命令如下：

```
python -m fenjing crack-request -f req.txt --host '127.0.0.1' --port 5000
```

Tab补全

参考这里配置shell启用tab补全

示例如下：

bash

```
cat >> ~/.bashrc << EOF
eval "$(_FENJING_COMPLETE=bash_source fenjing)"
EOF
```

zsh

```
cat >> ~/.zshrc << EOF
eval "$(_FENJING_COMPLETE=zsh_source fenjing)"
EOF
```

fish

```
echo '_FENJING_COMPLETE=fish_source fenjing | source' >
~/.config/fish/completions/fenjing.fish
```

注意只有输入`fenjing ...`的形式可以进行补全，`python -m fenjing`等形式无法进行tab补全

攻击失败怎么办？

如果payload生成失败，可以尝试调整以下选项：

- 使用`--detect-mode fast`减少请求次数，并优先使用更高级的绕过技巧
- 使用`--environment`手动指定目标的模板执行环境为flask或者jinja
- 使用`--waf-keyword`手动指定waf页面含有的关键字
- 使用`--detect-waf-keywords full`打开waf关键字检测功能

- 使用`--replaced-keyword-strategy`手动指定遇到字符替换型waf时的行为
- 使用`--eval-args-payload`减少请求次数