路由扫描

- python 3.8
- node 14.15.0
- MySql 8.0

登录功能

前端代码位置：scan_web/src/views/login/index.vue

后端代码位置：scan/views.py 中的 LoginView

因为这个系统主要是 扫描以及 漏洞，所以这里登录就简单

只要用户输入的账户密码都是 `admin` 即可认为是登录成功了。

用户信息

后端代码位置： scan/views.py 中的 UserInfoView

这里和上面一样，简略的处理。直接返回一个数据即可

退出登录功能

后端代码位置：scan/views.py 中的 LogoutView

和上面一样，简略处理。直接返回一个数据，告诉前端，退出成功。

扫描功能

前端代码：scan_web/src/views/form/index.vue

后端代码：scan/views.py 中的 ScanView

这里扫描用了 `Nmap` 和 `Apscheduler`

其中的 `Nmap` 是进行扫描的，这里就不再多说

`Apscheduler` 是用来处理扫描的。因为扫描的时候，用户会输入一个ip段，如果只用一个线程去扫描，会很慢很慢。这里的 `Apscheduler` 就是用来加速的，
把IP段分割为几个IP段，然后用它来处理。他就是相当于一个多线程处理的方式。也就是同时执行多个任务。
但是这个处理方式，不会阻塞主线程，也就是用户发起扫描请求后，后端会立马返回，不会一直holding住。

前端会有一个每2秒钟请求一次的检测接口，检测这次扫描是否完成。
如果完成了，就会显示画图，以及扫描出来的漏洞信息

扫描出的数据（品牌，型号这些），会用 Mysql 的全文检索功能，也就是一个相当于百度搜索的功能，只是数据集限定为我们抓取的CVE数据库，会在这里面匹配出漏洞，然后记录在数据库中。

下面就是建立全文检索的索引语句

爬取到足够多的数据后，创建 `fulltext` 索引(需要点时间)

```
ALTER TABLE cve_data ADD FULLTEXT INDEX cve_data_fulltext (description) with
parser ngram;
```

首页功能

前端代码：scan_web/src/views/dashboard/index.vue

后端代码：scan/views.py 中的 IndexView

这里就是把之前扫描的数据，汇总，给出画图的数据。

画图这里是用的 `Echarts` 这是百度开源的前端画图组件。

首页的功能，基本都是前端的代码比较多，都是用在画图上了，详细请看源代码

爬虫

- Scrapy

代码位置：cve_spider/cve_spider/spiders/cvedetails.py

爬虫采用scrapy进行编写。

这个数据量可以很多很多，我这里只爬取了5W来条数据。但是还可以爬的，不一定只能拿这么点数据。