概述

为了方便用户使用，我们直接将功能展示于web网页上。前端使用bootstrap、semantic、jquery、datatables和echarts等框架，后端主要是使用python2.7开发，使用flask框架，对json文件进行处理。

Flask是一个使用pyhton编写的轻量级 Web 应用框架。其 WSGI工具箱采用 Werkzeug ，模板引擎则使用 Jinja2 。相对于django框架，falsk更加灵活。本项目中主要使用了flask的@app.route()利用装饰器的方式实现了路由函数，方便读取json数据和传输json串给前端的ajax。

Datatables是jquery的表格插件，是一个高度灵活的工具，可以将任何HTML表格添加高级的交互功能。本项目中，利用Datatables实现威胁分析中的网站应用攻击（web容器、目录遍历、文件包含、文件上传、非法参数、黑客工具利用攻击、非正常HTTP请求、命令注入、SQL注入攻击、跨站脚本攻击（XSS））、系统层攻击（页面后门、系统漏洞、恶意病毒）、网络层攻击（暴力破解、CC攻击、防盗链）的数据展示，而且提供了按时间、类型等排序方式，还有搜索功能，可以轻松找到需要查看的数据，让用户一目了然。

Echarts是一个纯Javascript的图表库，可以流畅的运行在PC和移动设备上，底层依赖轻量级的Canvas类库ZRender，提供直观，生动，可交互，可高度个性化的数据可视化图表。本项目中的攻击总量、服务器攻击次数和攻击类型的图表，都是在echarts的基础上个性化出来的。将鼠标移动至每一个坐标点，将会显示详细信息，让用户能进一步了解到攻击状况。

Bootstrap和Semantic是HTML、CSS和JS框架，本项目中，Bootstrap提供页面的整体布局（如图），使页面简约而且兼容各种平台。使用Semantic实现页面一些模块的凸显风格，使页面看上去浑然一体。



在攻击事件详情菜单栏，使用jquery生成堆叠式菜单栏，不占空间，方便操作，同时，我们引入了iframe内联框架，嵌入了另一个页面，以弹窗的形式显示出来，附带动画过渡效果，给用户更好的体验。

对于数据处理，本项目中并没有使用数据库，因为我们在衡量数据库和json文件的性能时，经过测试结果显示，直接读取json文件的性能会更好，所以我们采用直接读取json文件的方式。将所有的json文件放在static/temp下，然后使用flask和jquery调用读取json文件显示在指定的标签中。Flask中先使用os.path.realpath读取json文件的真实路径，os.path.join对路径进行拼接，json.load将json文件加载到变量中，然后对变量值进行操作，最后通过jsonify方法将文件转换为json串的形式传递到ajax和前端标签中。

参考文献：

[1] Miguel Grinberg,Flask Web开发：基于Python的Web应用开发实战, 人民邮电出版社,2015:8-45