爬虫基本原理

黑客

黑客的定义

```
白帽黑客 --- 信息战场 ---破坏者
盾：防护                    矛：攻击


1. 白帽黑客：
        致力于检查系统的漏洞，防止恶意攻击
2. 破坏者：
        非法盗取信息，用于赚钱取材，获取敏感信息或资料，使系统瘫痪

3. 黑客的工作内容：
        1. 检测漏洞
        2. 入侵测试
        3. 安全管制
        4. 开发安全解决方案
        5. 安全咨询
        6. 安全培训
        7. 安全负责人
        8. 取证专家

4. 黑客攻击的用途：
        1. 应用程序黑客攻击：
                可以向运行中的应用程序插入任意的DLL或者源代码，拦截用户的键盘输入以盗取密码，可以插入图片，其他文件，网络传播
        2. web黑客攻击：
                可以创建网页爬虫，收集web页面包含的连接（手机app，pc，浏览器），实现SQL注入，向处理用户输入部分注入错误代码,使用Python可以实现浏览功能，通过操作htpp包，上传webshell攻击需要的文件
        3. 网络黑客攻击：
                可以实施网络踩点，搜索系统开发的端口，收集并分析网络数据包，进行网络嗅探，伪装无服务器，实施IP欺骗，非法获取敏感信息，可以大量发送数据包，实施拒绝服务式攻击
        4. 系统黑客攻击：
                可以编写后门程序，控制用户PC，开发用于搜索并修改PC注册的功能，可以利用程序的错误，通过缓冲溢出或者字符串实施攻击
```

- 为何要使用Python

python是黑客语言的首选
1. 支持强大的黑客攻击模块
    scrapy，pydbg，sqlmap，httplib
2. 可以访问各种API
    python可以使用其他语言的类库---胶水语言特性
    ctypes库--- windows（nt） OSX Linux Solaris FreeBSD OpenBSD
3. 简单，易上手
    黑客至少要掌握4门语言

网络爬虫的原理

引言：
    大数据（big data），爬虫的地位越来越高

- 数据获取方式

1. 第三方数据平台购买
    不合法---违法
    数据买卖，数据交易平台，数据不同，价格不同
2. 企业生产的用户数据：
    系统日志，用户日志
    用户数据
    中大型企业
3. 数据管理咨询公司
    有庞大的数据采集团队（合法）
    通过市场调研，调查问卷，样本检测，公司合作，专家对话
4. 政府/机构提供的公开数据
    政府通过各地的数据，进行统计合并，权威的数据
5. 爬虫自动获取的大量数据
    不用花钱，获取的数据有针对性，开发成本低（时间少，资源少）

- 什么是爬虫

定义：请求网站并提取数据的自动化程序

- 爬虫的用途

1. 作为大数据的数据源
2. 搜索引擎的数据来源
    百度--搜狗
    google

- 爬虫的类型

1. 通用网络爬虫：
    大型搜索引擎
    优点：数据量大，快速

缺点：有价值的数据较少
2. 聚焦爬虫：（企业最常用的爬虫）
也称之为主题爬虫
按照预先设定好的主题（格式），有选择的进行网页的爬取
优点：节省资源，数据价值较高
缺点：限制较大，通用性差
3. 增量式爬虫：
是一种特殊的聚焦爬虫，只采集更新后的数据
4. 深层网络爬虫：
抓取互联网中更深层次的数据
可能隐藏在了表单之下，不能通过静态的连接来获取，获取需要提交关键字之后才可获取

- 爬虫的作用范围

1. 只要能看的见，就一定能爬得到
2. 看不到的数据，能爬
构建网站进行采集（深层网络爬虫）

3. 爬虫的君子协定：
君子协定：口头协定，哪些能爬取，哪些不能爬
网站/robots.txt ---先查看云子协定
百度：
```
User-agent: *  # 所有身份的爬虫
Disallow: /    # 在/路径下不允许爬（全不允许）

Sitemap：目录     该目录允许大型浏览器访问（不允许私人访问）
```

- python做爬虫的优势

1. PHP
‘最好的语言’
不严谨
对于多线程，和异步支持不好，并发处理能力弱，速度和效率较低
2. Java
代码量较大，重构成本过大（维护和升级成本大）
3. C/C++
运行效率高，但是学习成本高，代码成型慢，开发周期长，不能应用于生产
面向过程的编程语言
4. Python
语言优雅，代码简洁，开发周期短，面向对象，类库多，代码量少，胶水语言，跨平台。。。

HTTP协议

- HTTP协议

```
1. HTTP协议，超文本传输协议：
        HyperText Transfer Protocol 超文本传输协议
        1. 是用于从www服务器传输文本到本地浏览器的一种传输协议
        2. 可以让浏览器更加高效，减少网络传输
        3. 保证计算机正确快速的传输文本文档
            确定传输的文档在哪一部分先显示（支持异步处理）
2. HTTPS 协议：
        加密的HTTP协议
        比HTTP更安全，作用一样
```

- HTTP主要请求方式

```
1. GET请求：
        最简单的请求，可以携带数据，不安全
        明文请求
        效率高，数据量有限制
        地址栏，表单提交手工设置，Ajax请求，超链接
2. POST请求：
        可以携带数据，更安全，加密
        密文请求
        效率低，没有数据量限制
        表单默认，Ajax请求
3. PUT请求：
        请求在服务器中存储一个资源，要指定一个存储位置
4. DELETE请求：
        请求在服务器中删除一个资源
5. HEAD请求：
        请求在服务器中的头信息
6. OPTIONS请求：
        可以获得当前URL多支持的请求类型（GET，POST）
7. CONNECT请求：
        HTTP/1.1协议中的预留给能够将连接改为管道方式的代理服务器
8. TRACE请求：
        回显服务器的请求，主要用于测试或诊断
```

- HTTP请求的7个步骤

```
1. 建立TCP/IP连接
2. web浏览器向web服务器发送请求命令
3. web浏览器发送请求头信息
        携带请求信息
4. web服务器响应（应答）
5. web服务器发送响应头文件
6. web服务器向浏览器发送数据
7. web服务器关闭TCP连接
```

- HTTP头部信息

1. HTTP头部信息的构成
    1. 头域
        域名：值
    2. headers：
        1. General：通用信息
        2. ResponseHeaders：响应头信息
        3. RequestHeaders：请求头信息

爬虫的原理

- 爬虫怎样从网页抓取数据

网页的三大特征：
1. 网页都有自己唯一的URL
2. 网页都是使用HTML描述页面信息
3. 网页都是使用HTTP/HTTPS协议传输数据

爬虫的设计思路：
1. 确定需要爬取的网页的URL
2. 通过HTTP协议或HTTPS协议访问服务器，拿到响应的页面
3. 提取HTML中的数据，并保存

- 爬虫的基本流程

1. 发送请求
    通过HTTP库向目标站点发送请求（request对象）
    请求可以包含额外的信息（cookies，user-agent，表单提交数据）
2. 获取响应的内容
    如果服务器响应正常，会得到一个响应对象（response对象）（HTTP库响应的）
    从响应中，获取指定的数据
    响应的内容：
    1. HTML页面
    2. JSON串
    3. 二进制数据（软件，音频，视频）
3. 解析响应
    得到想要的数据
    1. HTML页面：正则，xpath语法（xml解析）
    2. JSON串：直接对JSON解析　使用json模块
    3. 二进制文件：直接存储
4. 保存数据
    保存的数据，尽量转换成字符串

说明：请求和响应本质都是一个二进制流

- 请求

1. 请求的结构：
    post get
    携带数据：表单提交，url中的数据，headers，cookies
2. 请求的URL：
    URL：同一资源定位符
    定位一个html页面　定位一个二进制数据
3. 请求头：
    包含头部信息（cookies，表单的数据，url中的数据）
4. 请求体：
    请求时额外携带的数据

- 响应

1. 响应的状态：
    200：正常响应
    301：跳转
    404：资源未找到
    500：服务器错误
2. 响应头：
    内容的类型，内容的长度，服务器信息，cookies信息
3. 响应体：
    请求的具体资源，网页，图片，视频

数据的选择和处理

- 怎样抓取数据

1. 网页：
    正则，xpath
    re模块
    lxml模块
    bs4（BeautifulSoup）模块
2. JSON：
    json模块
3. 二进制数据：
    直接保存

- 解析方式

1. 直接处理
2. 正则表达式
3. JSON解析
4. xml解析相关模块
5. 解决JS渲染
    1. 一切响应以http包响应的内容为准
    2. Splash模块
    3. pyV8模块

爬虫和反爬虫

- 反爬虫

```
1. 网站，可以识别用户身份
        User-Agent  识别身份
        1. 如果不是浏览器，被识别为机器人（爬虫程序），进制访问（封账号，封IP）
```

- 常见的反爬虫策略

```
1. 通过分析Headers信息进行反爬虫
        最长见，使用的最多
2. 通过验证用户行为反爬虫
        如果短时间内，多次访问同一个网站，可能会触发反爬虫机制
3. 通过动态页面增加爬虫爬取的难度，达到反爬虫的目的
        智联招聘：至少有5中模板

应对策略：
1. 构建Headers信息，伪装成浏览器
2. 1.换IP，继续获取数据  2.伪装成人的行为，攻破技术关，继续爬取
3. 编辑多个模板，应对难度的增加
4. 利用无界面浏览器
        selenium

注意：
1. 反爬虫不会过于严格，可能会误伤正常用户
2. 增加研发成本，所以反爬虫不会太严格
3. 伪装越少，效率越高，但是容易被检测
4. 高度伪装，效率低，但是不易被识别
```

- 补充 Headers的头信息

```
1. General
        1. Request URL： 资源url
        2. Request Method：请求方式
        3. Status Code： 响应的状态码
        4. Referrer Policy：认证信息（用于反爬虫）
2. ResponseHeaders
        1. Connection：连接方式
        2. Content-Encoding：设置内容编码
        3. Content-Type： 内容的类型
        4. Date：日期
3. RequestHeaders
        Accept:客户端能接收的资源类型
        Accept-Encoding:设置编码的格式（压缩流）
        Accept-Language:语言
        Cache-Control:缓冲控制，0：表示进制缓存
        Cookie: 用于存储用户信息（常用于登录状态保持）
        Host: 连接目标主机的IP和端口
        Referer:请求的数据源-上一个请求来自于哪里
```

```
        User-Agent:识别用户信息（操作系统，浏览器版本号）
```

- 总结

```
反爬虫的手段有很多，但是都有明确的对应方式
最常用的反爬虫手段：User-Agent
```
