# 危险情报自动收集软件工程设计思维图

## 主要核心模块化功能：

1 备份文件扫描

2 cms指纹识别

3 使用struts\_2框架搭建的网站 -- 删除

4 sql注入简单检测

## 1 备份文件扫描

### 扫描项目

A扫描备份文件，后缀为网站的域名加上常见的后缀名称['.rar','.zip','.tar','.tar.bz2','.sql','.7z','.bak','.txt']

技术实现：用字符串切割方式组合域名加备份文件名称 .split

比如[www.baidu.com/baidu.rar](http://www.baidu.com/baidu.rar) [www.baidu.com/baidu.zip](http://www.baidu.com/baidu.zip)

为什么要这么做呢，因为很多网站在备份的时候备份文件名称是用域名作为名称

B

扫描备份文件，后缀为固定的['root.rar','beifen.rar'....等等]

技术实现：读取本地的常用备份文件txt，转换成列表，字符串拼接

比如 www.baidu.com/root.rar

C

扫描常见的源码泄露漏洞,包括svn源码泄露，git源码泄露

技术实现：字符串拼接 加上 /.svn/entries /.git/config

比如 www.baidu.com/.svn/entries

### 技术思维

使用requests库实现url请求，采用head访问方式，在得到的repsone中返回的对象是字典形式，可以采用["Content-Length"]读取返回内容的大小，一般来说大小都会大于1M也就是1048576个字节，只要验证大小即可

同理，svn和git源码泄露，由于返回的内容比较小，可以采用get访问方式，在返回的页面中寻找相关特征码得以实现。

详细细节

A 中 backup\_name\_A = ['.rar','.zip','.tar','.tar.bz2','.sql','.7z','.bak','.txt']

比如 urlx = <http://www.baidu.com>

那么url = urlx + '/' + urlx.split(".",2)[1] >>>><http://www.baidu.com/baidu>

将url带入到backup\_name 中迭代一次 循环访问就是A技术的实现手段

B中可以采用直接读取文本字典的形式，有一个rar.txt,其中的内容是这样的



将这份文件逐行读取，然后组合成列表

Backup\_name\_B=list(set([i.replace("\n","") for i in open("rar.txt","r").readlines()]))

Set的作用是把一个列表转换成集合，因为集合中不能有两个相同的数据，所以能达到去重复的效果，去重复之后再把这个集合转换成列表的形式，集合和列表之间互相转换的方法就是直接set(对象) list(对象)就可以了

然后把url加载进去迭代循环使用requests访问，判断返回对象大小即可

需要注意的是，由于有些网站并不能扫描到备份文件，所有网址返回的headers头部信息中并没有Content-Length（文件大小），这个时候的话就会出问题，可以使用try except处理异常机制

C 中 可以直接使用url + /.svn/entries /.git/config 组合成url地址

然后用requests.get直接访问 判断其中的关键词，比如SVN泄露，这个页面中会有‘dir’ 和’svn’这两个单词，可以使用if ‘ svn’ and ‘dir’ in r.content方式判断

Git同理

url\_svn 是svn泄露链接

url\_git 是git源码泄露

## 2 cms指纹识别

验证方式有三种

1 直接访问页面 找寻特种cms特定的关键字

2 访问网站的robots.txt 在这个txt寻找特定的关键字

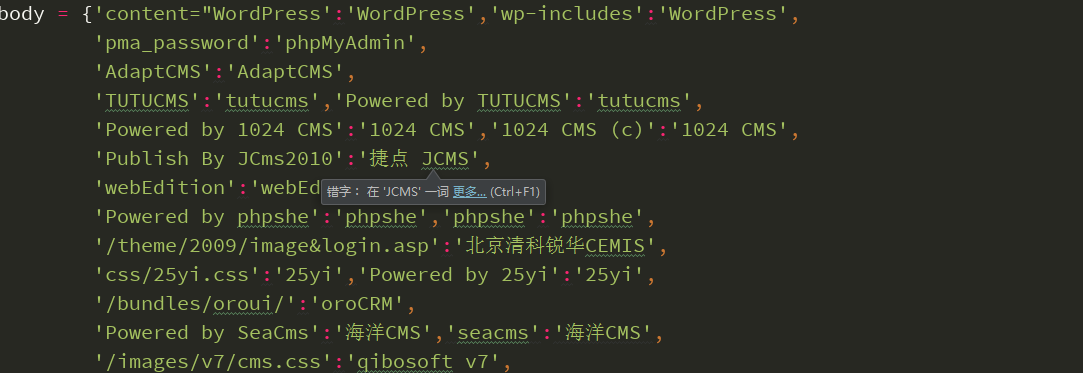
3 cms指纹

技术实现

【需要注意的是 防止黑产泛滥引发的一系列思考】

1 用字典的形式加载关键词和cms

比如



字典的Key是关键词 vaule是cms类型 关键词出现在网页的首页文件里面

So 就像检测svn源码泄露一样 get网站，然后寻找关键词就可以

2 访问网址拼接/robots.txt 然后查找关键字 同上

3 写入指纹库 然后验证指纹 详见代码

使用st2框架的网站