多维度指纹规则识别CMS

作者: LANDGREY ● 创建时间 2017年5月30日 17:13 ● 更新时间 2017年5月30日 22:08
浏览: 1228 次. ● 标签: #渗透测试
您的IP地址: 140.207.23.83

0x00： 前言

CMS(Content Management System)类型的识别是老生长谈的问题，这篇文章就当抛砖引玉了~

识别CMS类型的主要方式:

1. 根据网页内容含有的特殊表征
2. 根据网站路径的确定静态资源
3. 根据网站的HTTP响应头
4. 根据网站目录结构的整体特征

用的最多的，就是第一种和第二种探测方式；第三种通用性较差，不太常用；第四种几乎没见到过具体实现，因为不实用，耗时长，而且容易被waf屏蔽。

0x01： 识别方式简介

1. 根据网页内容含有的特殊表征

```
网站主页内容含有的一些特殊字符串，如Powered by xxxcms；
引用css特殊的标志，如 Metinfo 主页常含有class="met-navfixed" 、class="met_clear" 等CSS引用标志；
主页内容含有的特殊的正则表达式，如 PHPcms 主页常有类似于正则 /index\.php\?m=content&c=index&a=lists&catid= 的字符
```

2. 根据网站路径的确定静态资源

```
robots.txt 中含有的cms名称等特殊字符串；
favicon.ico 图标的hash值；
某个特殊路径的图片、css文件、README文件、license文件、甚至文档等资源的hash值或特殊字符串；
```

3. 根据网站的HTTP头

```
某个路径的图片、css等文件存在的status code： 200 响应；
ThinkPHP 常爆出带'哭脸'的"无法加载控制器"等错误响应；
另外，还可以通过网站HTTP headers、Cookies等字段含有的特殊值来判断，不过不常用，也没必要；
```

4. 根据网站目录结构的整体特征

```
需要爬取网站，基本不会单独实现此功能
```

0x02： 程序构想

单个目标，基于多维度规则，设定并判断权重，多线程运行， 网站响应内容复用，得分判断CMS类型。

为了方便加载和统一设置规则，选择使用正则表达式解析规则文本。一种CMS规则存取到一个文本文件中，例如一个检测discuz!7.x版本的规则如下：

```
type:           [discuz!7.x]
help:           []
content:        ['content="Discuz! 7\."', 'content="2001-2009 Comsenz Inc\."']
regex:          []
statics:        ['/images/smilies/default/biggrin.gif': 'FD4CD2A1F608C189BFDD3DADD0EB28C3']
special:        []
codeStatus:     []
```

如果后期需要增加判断的一个维度，只需在最后一行增加如：

```
newrule:        [xxx]
```

并修改相应的正则表达式、添加检测逻辑即可。每个CMS规则文件中的维度均可选择填写一至多种，方便将他人的规则吸收进自己的规则中，实现"大一统"。

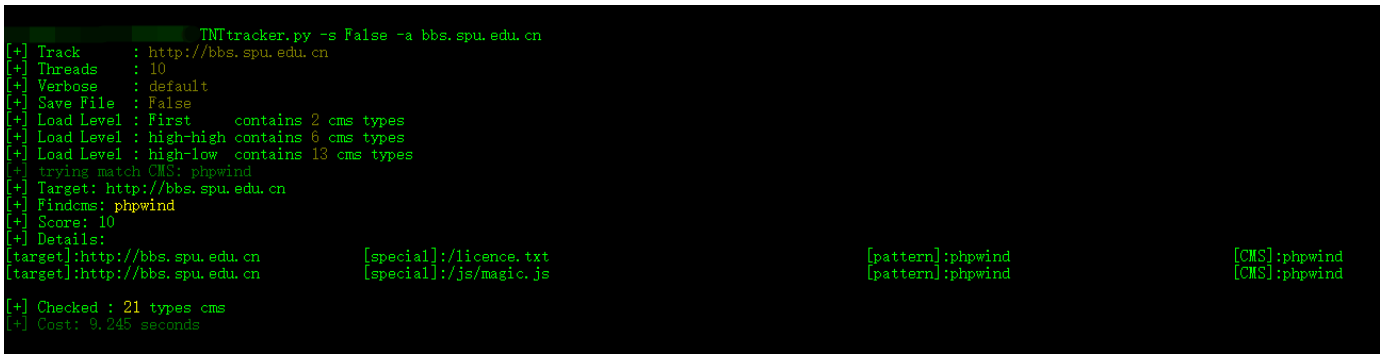
每种维度设定合适的权重，每命中一条，增加相应的权重分数，到达一定分数即可认为确定CMS类型，输出相关信息，然后退出。

程序可以设置一个字典，存储网站对某个请求路径的响应，如果有重复的请求路径，可以直接从内存读取，不必再次从网站请求，防止请求次数过多被屏蔽。

另外，还可以根据CMS的流行程度和使用量，划定探测CMS的优先顺序，争取更快的探测出CMS类型。

0x03：代码实现

目前初步完成程序，TNTtracker 的实际使用效果如下图所示：



程序暂时不会开源，所以不能展示相关代码。

0x04：效果分析

与其它CMS识别系统一样，TNTtracker的识别能力也是基于**规则的准确度**和**规则集的数量**；

优势：

1. 识别维度增加，可以选择性的填写相关识别维度，识别CMS，也方便吸收其它的识别系统规则；
2. 使用正则表达式获取规则集，可增加维度，提升识别能力，适当修改，可吸收其它检测方式的规则；
3. 可根据CMS的流行程度和使用数量，选择优先识别的CMS类型，提高识别速度；
4. 根据不同的维度权重，命中判分识别，比一刀切判断更加灵活；
5. 已经请求过的路径保存到内存中，需要再次使用时从内存读取，防止多次请求；

劣势：

1. 没有摆脱基于规则指纹的识别方式，仍是老一套；
2. 和老一套一样，没有标准的规则和要使用的维度，识别效果一定程度上依赖选定规则的人；
3. 程序没有做具体的版本识别，实在要添加的话只能通过不同的规则文件来实现；

0x05：后记

比较了各种维度的检测方式后，个人还是倾向于先**使用主页的正则表达式模式和使用静态文件的MD5值来判别**CMS：

一是不容易修改，比较准确；

二是一些二次开发的CMS可能依然保留一些正则特征和原CMS的一些静态文件；

三是一定程度上可以降低大量请求触发waf的概率。

添加规则过程中发现现有公开的规则质量堪忧，一些二次开发的CMS检测出的几率很小，去掉些banner和特征之类的可能就检测不出来了，别说一些深度改装的CMS了。

其它还可以**增加通用和非通用目录的cms探测**，比如网站主站是自己开发的，但是在"/wordpress"目录下存在一个wordpress系统，如果没有探测到的话，就可惜了。

<