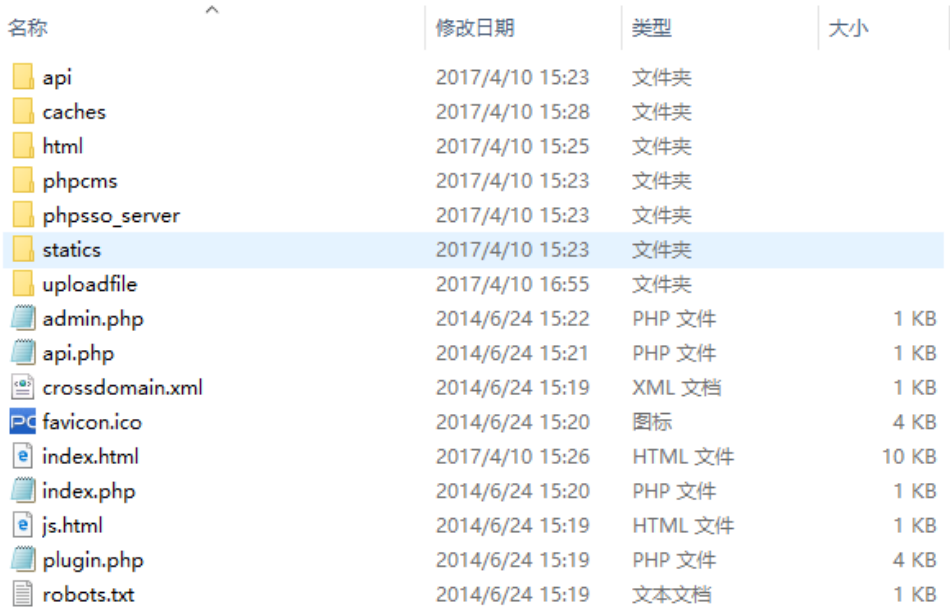
审计思路之实例解说全文通读

根据敏感关键字来回溯传入的参数，是一种逆向追踪的思路，我们也提到了这种方式的优缺点，实际上在需要快速寻找漏洞的情况下用回溯参数的方式是非常有效的，但这种方式并不适合运用在企业中做安全运营时的场景，在企业中做自身产品的代码审计时，我们需要了解整个应用的业务逻辑，才能挖掘到更多更有价值的漏洞。

全文通读代码也有一定的技巧，并不是随便找文件一个个读完就可以了，这样你是很难真正读懂这套 Web 程序的，也很难理解代码的业务逻辑，首先我们要看程序的大体代码结构，如主目录有哪些文件，模块目录有哪些文件，插件目录有哪些文件，除了关注有哪些文件，还要注意文件的大小、创建时间。我们根据这些文件的命名就可以大致知道这个程序实现了哪些功能，核心文件是哪些，如下是 phpcms 的程序主目录。如图所示。



名称	修改日期	类型	大小
api	2017/4/10 15:23	文件夹	
caches	2017/4/10 15:28	文件夹	
html	2017/4/10 15:25	文件夹	
phpcms	2017/4/10 15:23	文件夹	
phpsso_server	2017/4/10 15:23	文件夹	
statics	2017/4/10 15:23	文件夹	
uploadfile	2017/4/10 16:55	文件夹	
admin.php	2014/6/24 15:22	PHP 文件	1 KB
api.php	2014/6/24 15:21	PHP 文件	1 KB
crossdomain.xml	2014/6/24 15:19	XML 文档	1 KB
favicon.ico	2014/6/24 15:20	图标	4 KB
index.html	2017/4/10 15:26	HTML 文件	10 KB
index.php	2014/6/24 15:20	PHP 文件	1 KB
js.html	2014/6/24 15:19	HTML 文件	1 KB
plugin.php	2014/6/24 15:19	PHP 文件	4 KB
robots.txt	2014/6/24 15:19	文本文档	1 KB

在看程序目录结构的时候，我们要特别注意几个文件，分别如下：

1) 函数集文件，通常命名中包含 functions 或者 common 等关键字，这些文件里面是一些公共的函数，提供给其他文件统一调用，所以大多数文件都会在文件头部包含到它们，寻找这些文件一个非常好用的技巧就是去打开 index.PHP 或者一些功能性文件，在头部一般都能找到。

2) 配置文件，通常命名里面包括 config 这个关键字，配置文件包括 Web 程序运行必须的功能性配置选项以及数据库等配置信息，从这个文件里面可以了解程序的小部分功能，另外看这个文件的时候注意观察配置文件中参数值是用单引号还是用的双引号包起来，如果是双引号，则很大可能会存在代码执行漏洞，例如下面 kuwebs 的代码，只要我们在修改配置的时候利用 PHP 可变变量的特性即可执行代码。

```
    <?php
    /*网站基本信息配置*/
    $kuWebsiteURL      = "http://www.kuwebs.com";
    $kuWebsiteSupportEn        = "1";
    $kuWebsiteSupportSimplifiedOrTraditional         = "0";
    $kuWebsiteDefauleIndexLanguage                  = "cn";
    $kuWebsiteUploadFileMax                        = "2";
    $kuWebsiteAllowUploadFileFormat   = "swf|rar|jpg|zip|gi
f";


    /*邮件设置*/
    $kuWebsiteMailType       = "1";
    $kuWebsiteMailSmtpHost           = "smtp.qq.com";
```

3) 安全过滤文件，安全过滤文件对我们做代码审计至关重要，关系到我们挖掘到的可疑点能不能利用，通常命名中有 filter、safe、check 等关键字，这类文件主要是对参数进行过滤，比较常见的是针对 SQL 注入和 XSS 过滤，还有文件路径、执行的系统命令的参数，其他的则相对少见。而目前大多数应用都会在程序的入口循环对所有参数使用 addslashes()函数进行过滤。

```
    private static function _do_query_safe($sql) {

                $sql = str_replace(array('\\\\', '\\\'', '\
\"', '\'\''), '', $sql);

                $mark = $clean = '';

                if (strpos($sql, '/') === false && strpos
($sql, '#') === false && strpos($sql, '-- ') === false && s
trpos($sql, '@') === false && strpos($sql, '`') === false)
{

                      $clean = preg_replace("/'(.+?)'/s", '
', $sql);

                } else {
```

4) index 文件 ,index 是一个程序的入口文件 ,所以通常我们只要读一遍 index 文件就可以大致的了解整个程序的架构，运行的流程，包含到的文件，其中核心的文件又有哪些，而不同目录的 index 文件也有不同的实现方式，建议最好是先把几个核心目录的 index 文件都简单读一遍。

上面介绍了我们应该注意的部分文件，可以帮助我们更有方向的去读全部的代码，实际上在我们真正做的代码审计的时候，经常会遇到各种框架，这时候就会被搞的晕头转向 ,所以在学习代码审计的前期建议不要去读开源框架或者使用开源框架的应用，先去 chinaz、admin5 一类的源码下载网站下载一些小应用来读一下，并且一定要多找几套程序通读全文代码，这样我们才能总结经验，等

总结了一定的经验，会 PHP 也比较熟悉的时候，再去读一些像 thinkphp、Yii、Zend Framework 等开源框架，才能快速的挖掘高质量的漏洞。

通读全文代码的好处显而易见，可以更好的了解程序的架构以及业务逻辑，能够挖掘到更多更高质量的逻辑漏洞，一般老手会比较喜欢这种方式。而缺点就是花费的时间比较多，如果程序比较大，读起来也会比较累。