目录

一、题目

1、源码

```php
class Redirect {
    private $websiteHost = 'www.example.com';

    private function setHeaders($url) {
        $url = urldecode($url);
        header("Location: $url");
    }

    public function startRedirect($params) {
        $parts = explode('/', $_SERVER['PHP_SELF']);
        $baseFile = end($parts);
        $url = sprintf(
            "%s?%s",
            $baseFile,
            http_build_query($params)
        );
        $this->setHeaders($url);
    }
}

if ($_GET['redirect']) {
    (new Redirect())->startRedirect($_GET['params']);
}
```

2、知识点

知识点	说明
end()	将内部指针指向数组中的最后一个元素，并输出
http_build_query()	生成URL-Encode之后的请求字符串
$_SERVER['PHP_SELF']	表示当前PHP文件相对于网站根目录的位置地址

3、解读

1）第21行，如果GET方法接收到 'redirect' 的值，就实例化Redirect对象，调用对象的函数startRedirect()，使用GET方法中 'params' 的值作为实参。
2）第9-19行，将 params 的值放在PHP文件位置进行拼接成url，并进行URL编码，作为函数setHeaders()的实参并调用。
3）第4-7行，对url进行解码并设置为跳转的地址。

4、分析

1）$_SERVER['PHP_SELF']将URL中的文件位置提取出来并作为下一个跳转的地址，那么这里变量就可控了。构造成其他网站URL，即可造成任意URL跳转攻击。
2）由于URL中的字符在进入到服务器之后，本身会先进行一次URL解码，然后第5行，又对URL进行了一次解码，所以这里考虑URL双编码的形式构造Payload。

5、利用

```
http://www.xxx.com/index.php/http:%252f%252fbaidu.com?redirect=test&params=test
```

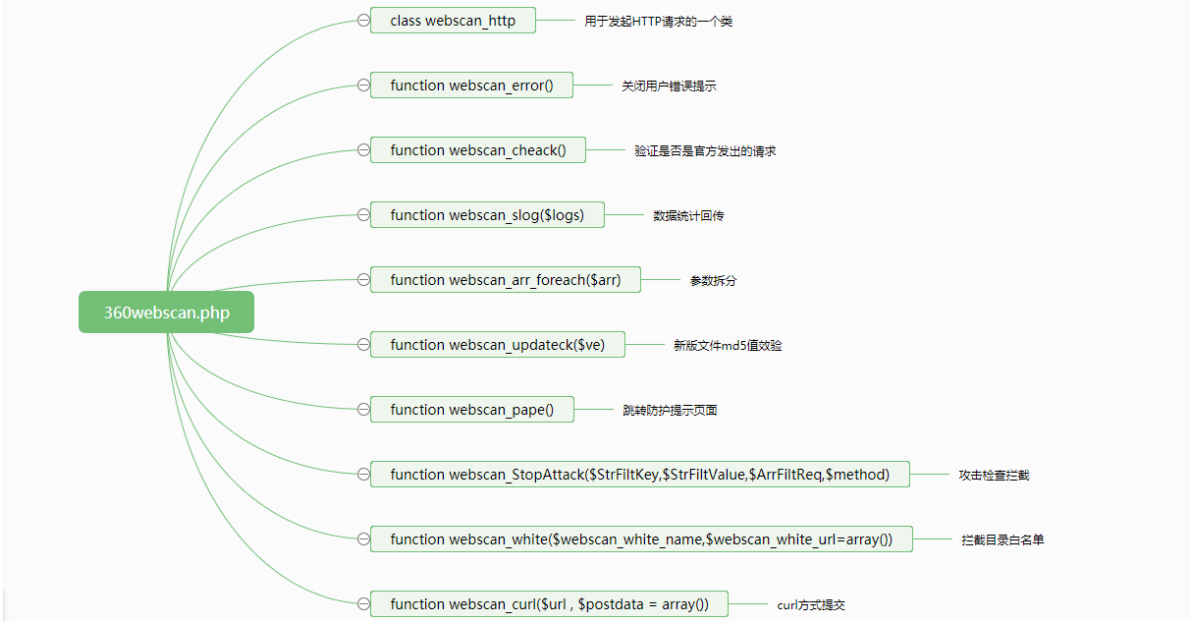
二、CMS-360webscan

1、知识点

Null

2、分析

1）360webscan是一个防护脚本，也就是可以配合任意CMS进行使用，对网站起到保护作用。



2）那么此时如果输入敏感字符，如 <script>alert(1)</script>，就会被拦截

3）回到360webscan脚本中，可以看到拦截目录白名单，其中第5行，通过$_SERVER['PHP_SELF']接收值并作为$url_path的值，这里并没有做过滤。

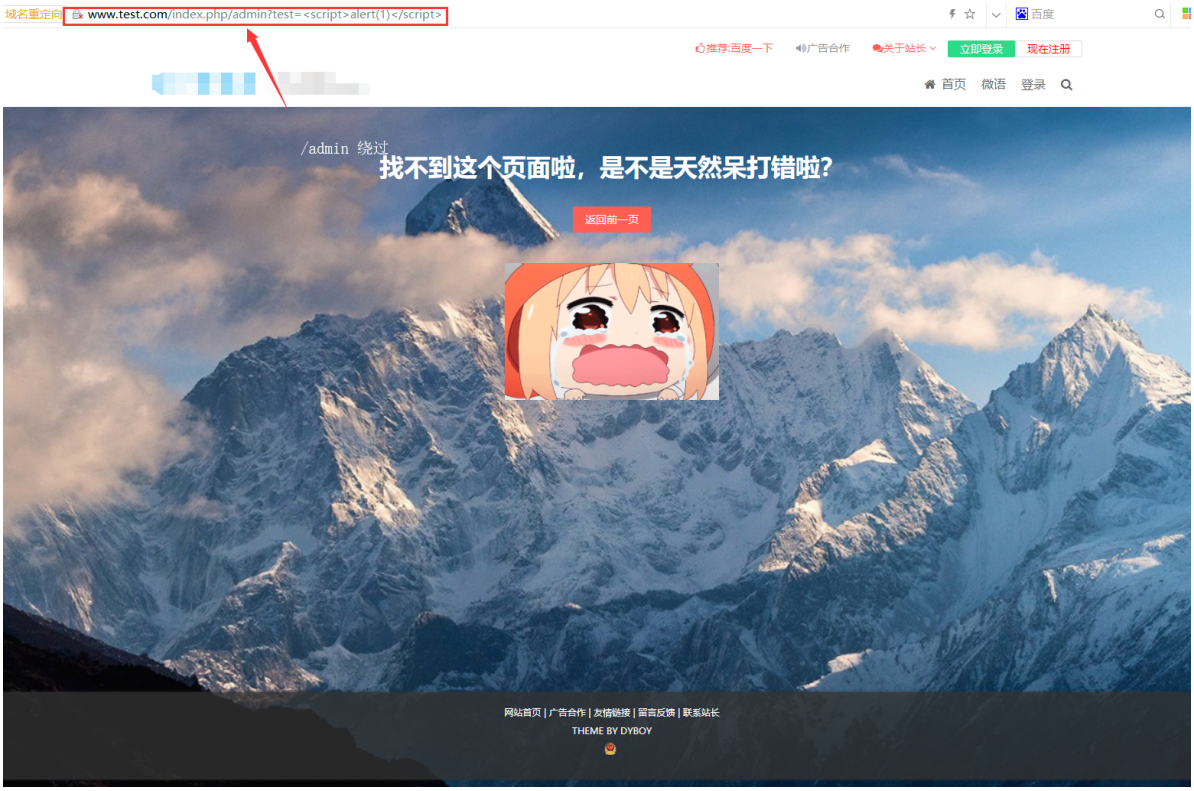
```php
/**
 *  拦截目录白名单
 */
function webscan_white($webscan_white_name,$webscan_white_url=array()) {
  $url_path=$_SERVER['PHP_SELF'];
  $url_var=$_SERVER['QUERY_STRING'];
  if (preg_match("/".$webscan_white_name."/is",$url_path)==1&&!empty($webscan_white_name)) {
    return false;
  }
  foreach ($webscan_white_url as $key => $value) {
    if(!empty($url_var)&&!empty($value)){
      if (stristr($url_path,$key)&&stristr($url_var,$value)) {
        return false;
      }
    }
    elseif (empty($url_var)&&empty($value)) {
      if (stristr($url_path,$key)) {
        return false;
      }
    }

  }

  return true;
}
```

4）然后在webscan_cache.php中，白名单目录中存在admin 和 /dede/，对这两个目录文件是放行的。

```php
<?php
//用户唯一key
define('WEBSCAN_U_KEY', '6809abbda8d53816f11500b52637e8db');
//数据回调统计地址
define('WEBSCAN_API_LOG', 'http://safe.webscan.360.cn/papi/log/?key='.WEBSCAN_U_KEY);
//版本更新地址
define('WEBSCAN_UPDATE_FILE','http://safe.webscan.360.cn/papi/update/?key='.WEBSCAN_U_KEY);
//拦截开关(1为开启，0关闭)
$webscan_switch=1;
//提交方式拦截(1开启拦截,0关闭拦截,post,get,cookie,referre选择需要拦截的方式)
$webscan_post=1;
$webscan_get=1;
$webscan_cookie=1;
$webscan_referre=1;
//后台白名单,后台操作将不会拦截,添加"|"隔开白名单目录下面默认是网址带 admin  /dede/ 放行
$webscan_white_directory='admin|\/dede\/';
//url白名单,可以自定义添加url白名单,默认是对phpcms的后台url放行
//写法：比如phpcms 后台操作url index.php?m=admin php168的文章提交链接post.php?job=postnew&step=post ,dedecms 空间设置edit_space_info.php
$webscan_white_url = array('index.php' => 'm=admin','post.php' =>
'job=postnew&step=post','edit_space_info.php'=>'');
?>
```

3、利用

网站中使用admin或者/dede/目录，再次传入 <script>alert(1)</script>，则不会被拦截。也就是说，使用这个360webscan脚本的网站，都是可以在 admin 和 /dede/ 这两个目录文件进行敏感字符操作的，使用白名单绕过了防护。



4、修复方案

使用 $_SERVER['SCRIPT_NAME']代替 $_SERVER['PHP_SELF']。

5、参考链接

https://github.com/hongriSec/PHP-Audit-Labs/blob/master/Part1/Day15/files/README.md