奇安信攻防社区 – 记一次 SEMCMS 代码审计

奇安信攻防社区 – 记一次 SEMCMS 代码审计

用 Seay 源代码审计系统扫描出来的还有 XSS 注入漏洞，还有文件包含漏洞，文件包含漏洞用函数 include_once 调用文件，但是没有用参数来获取这个调用，可能不存在文件包含漏洞。xss 注入漏洞因为过滤了 < 符号，在构造的时候可能会被过滤掉。

前言

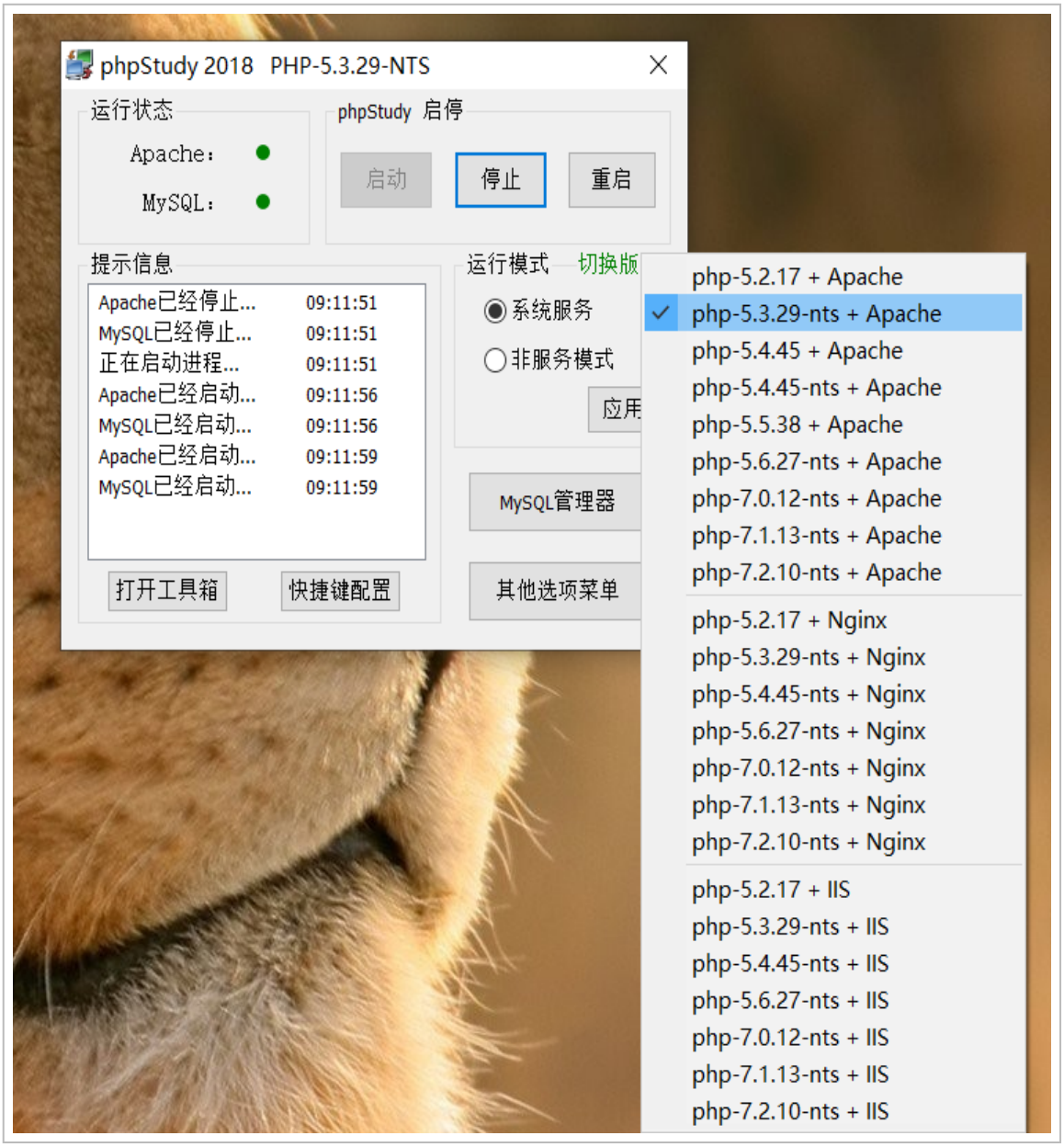
第一次做代码审计，有些地方没有分析得很透彻，请见谅！当然该 cms 还存在一些其他的漏洞，感兴趣的同学可以去研究一下。

一、搭建环境

首先，在 http://www.sem-cms.com/xiazai.html (http://www.sem-cms.com/xiazai.html) 网站上下载 SEMCMS（V3.9 版本）的源代码，将其放下 phpstudy 软件下，就能够搭建起一个 web 环境。

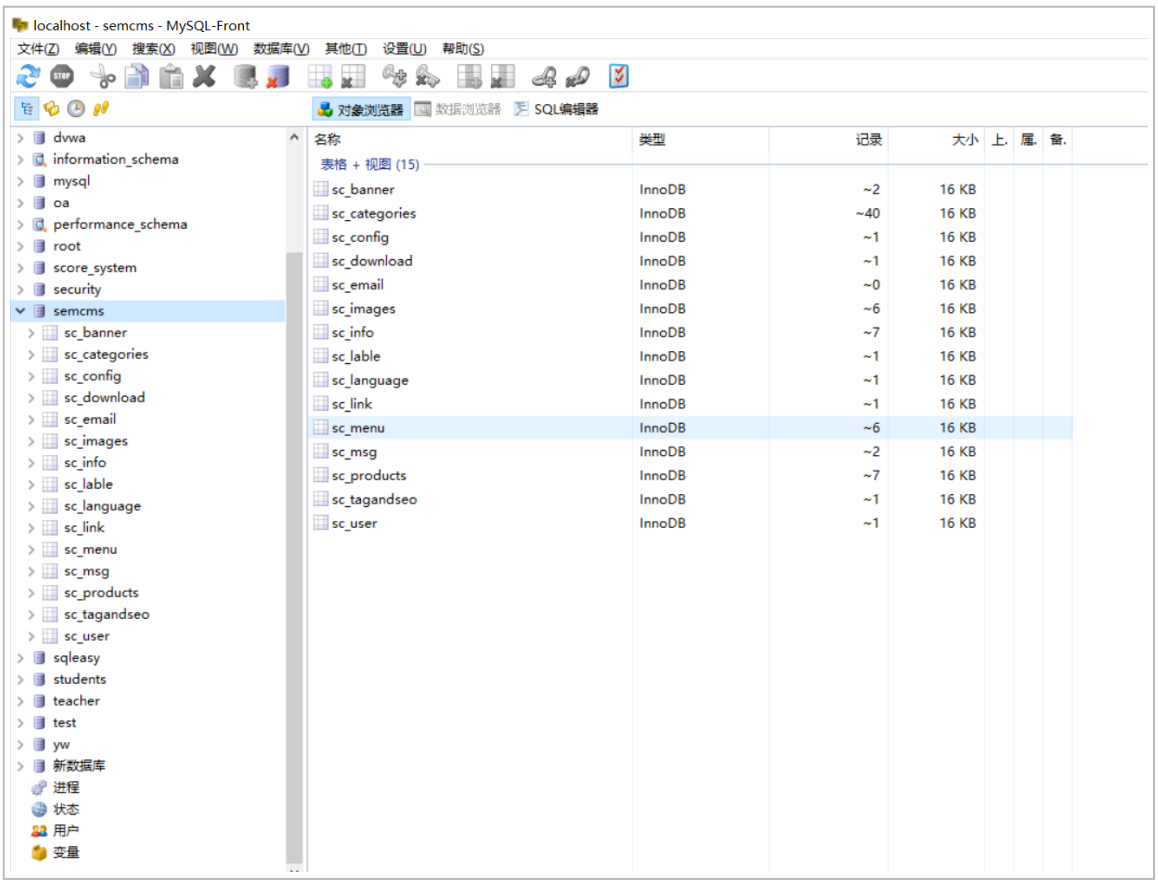
(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-2d1b4c58289a13d498b40a4dce51257f7679b811.png)

设置一个中间件的版本号



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-d55b4508d866ff680e5ce2a9fe84dce7a2352afd.png)

将 semcms.sql 数据包导入数据库中



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-c03c253c5581d3c9f4009dc108982460791d5ec9.png)

二、代码审计

源码如下，分为如下几个部分

/A_Admin 目录下是后台界面管理相关的文件

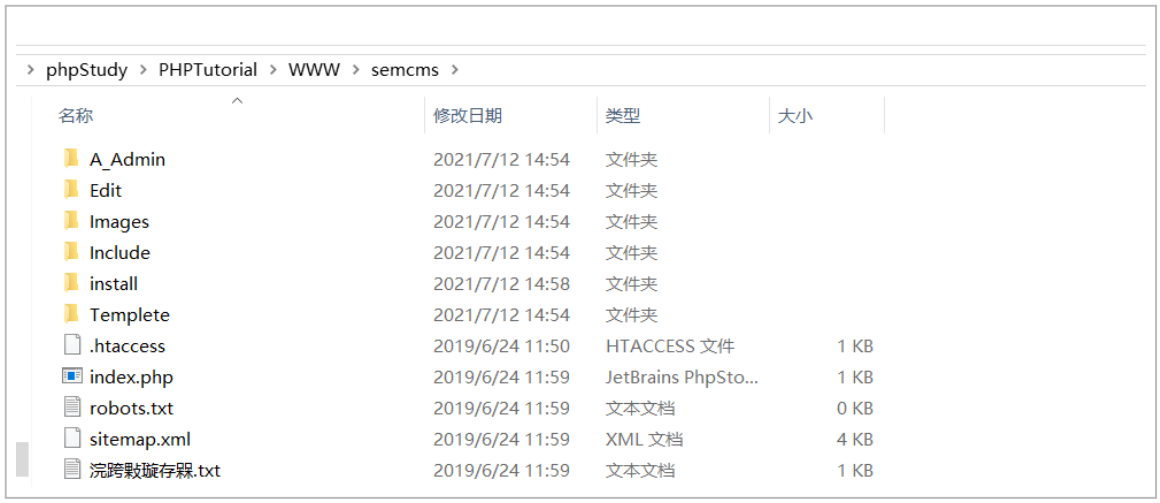
/Edit 目录下存放的的是 js 配置文件和 html 配置文件

/Image 目录下存放的是网页中的各种图片

/include 目录下存放的是包含全局的文件

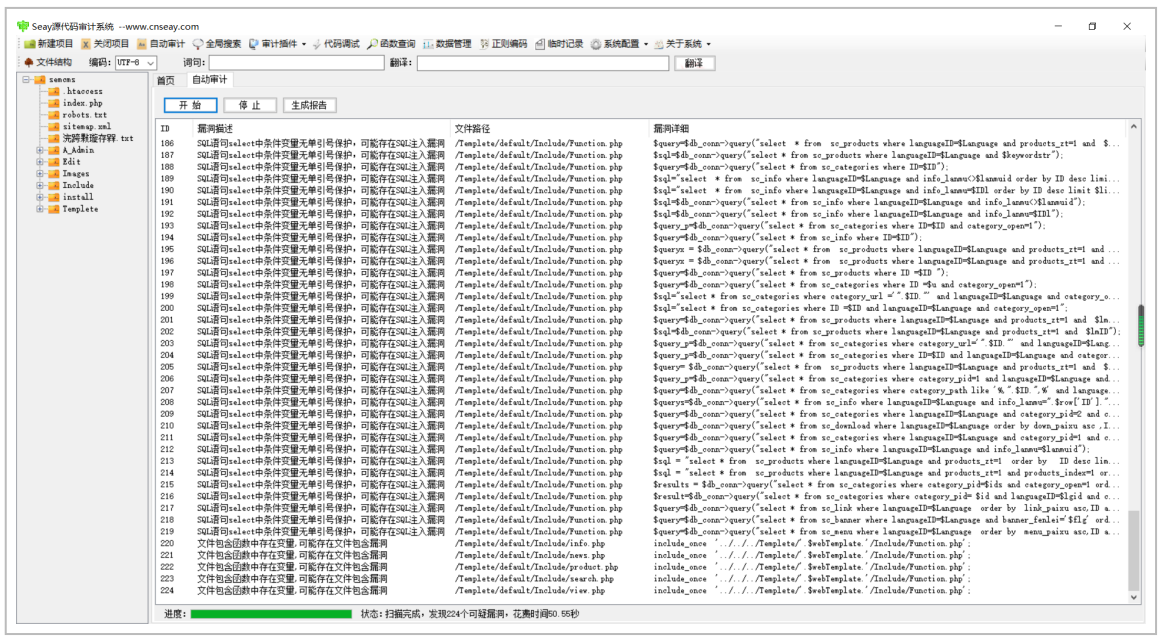
/install 目录下存放的是 semcms.sql 数据库备份文件

/Templete 是一些 js 配置文件和网站配置文件



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-e5c066d1cb3dc9e39a0c546cef86edba514b26a7.png)

将该 cms 源文件放在 Seay 源代码审计系统进行审计



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-910ef544a6572f492451f627a431fdae5d9f3b20.png)

在源码中的 `/Include/contorl.php` 的第 5 行 – 第 26 行中将所有 GET 请求中的

传参过滤，同时定义了 `verify_str()函数` ，当调用这个函数时，会调用 `inject_check_sql($str)函数` 对字符串进行正则匹配，当匹配成功时，执行 exit('Sorry,You do this is wrong! (.-.)'); 匹配不成功则返回字符串。

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-d34e94cb2671e4aaa6bc530d27d9d0609474f4fd.png)

preg_match（）函数分析

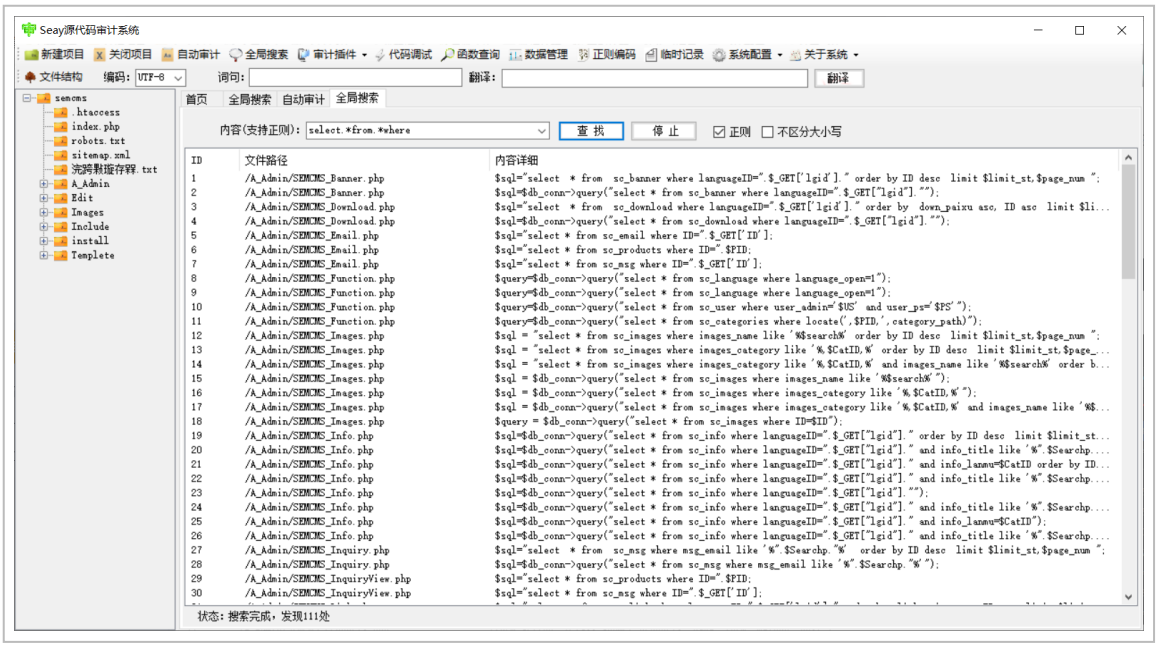
1. 末尾的 / i 表示对大小写不敏感
2. 匹配 $sql_str 参数里的有关的参数，从下面的过滤条件可知过滤了常用的 sql 语句 select、union、单引号闭合、*、= 以及一些其他的 SQL 语句
3. 所以如果想要构造 SQL 注入，就要找到双引号闭合的或者是数字型注入的点，或者是以 post 方式传参且没有对单引号闭合的注入点。

执行 SQL 语句的过滤

`return preg_match('/select|insert|=|%|<|between|update|\'|\*|union|into|load_file|outfile/i',$sql_str);`

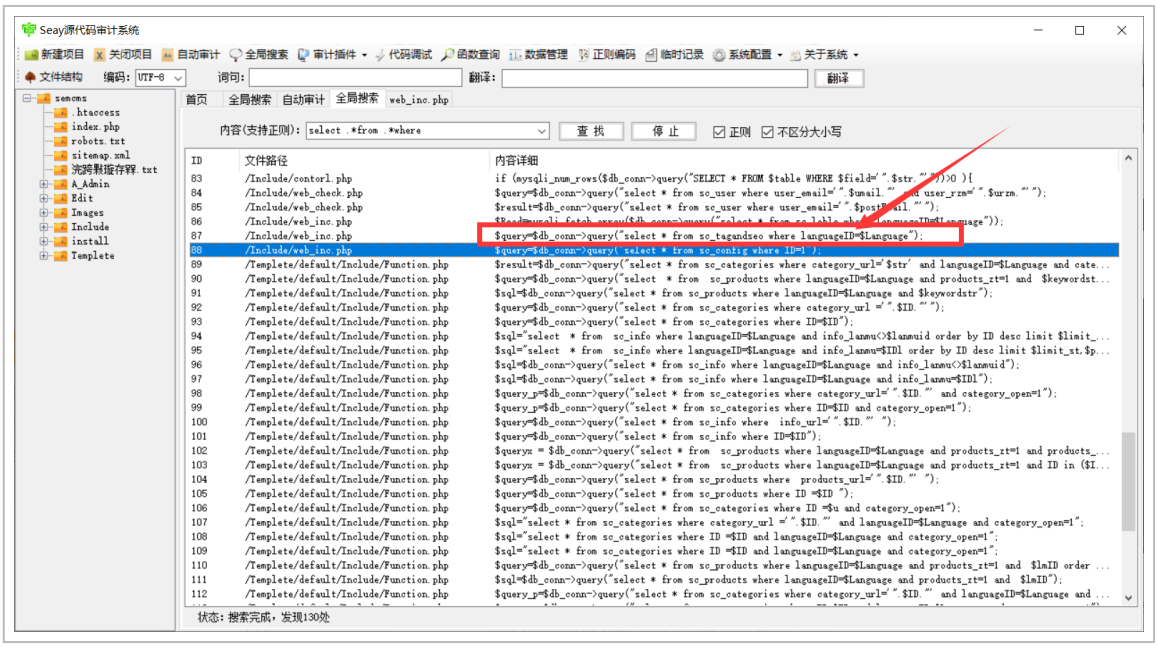
通过 seay 源代码审计系统对文件进行一个全局搜索

`Select.*from.*where`



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-a6fc96524a3468725a0aa481c6ddfe0d2c4a17c8.png)

通过全局搜索找到了一条数字型 SQL 注入，通过 $language 来传参

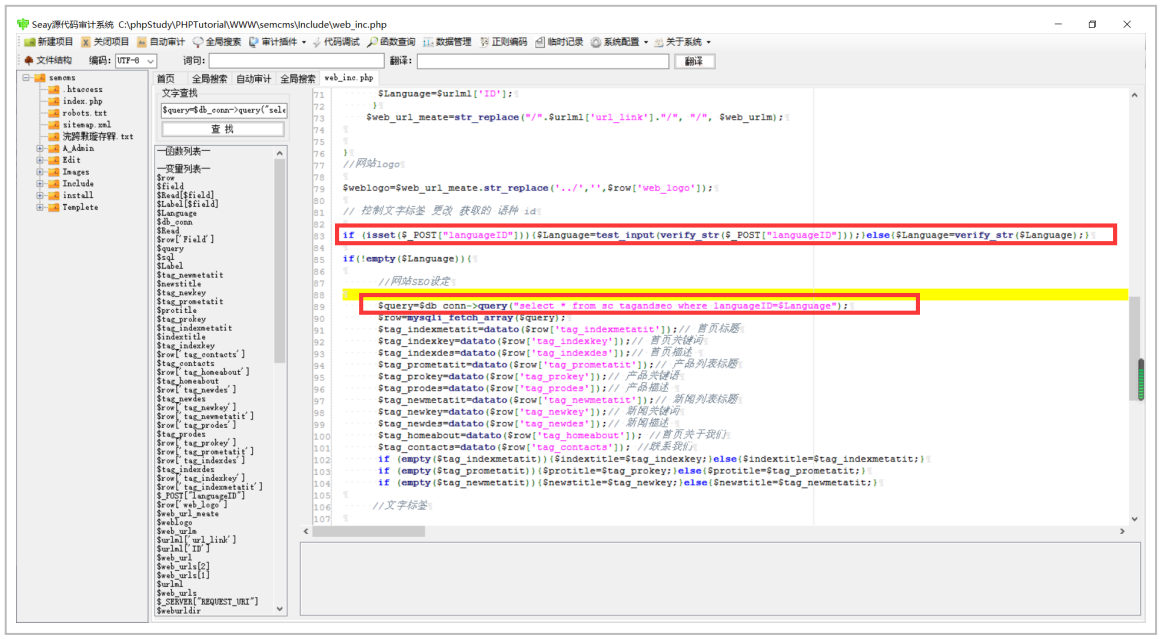


(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-77b653ad1f7058e356899f98ed07a6e8d06f68c6.png)

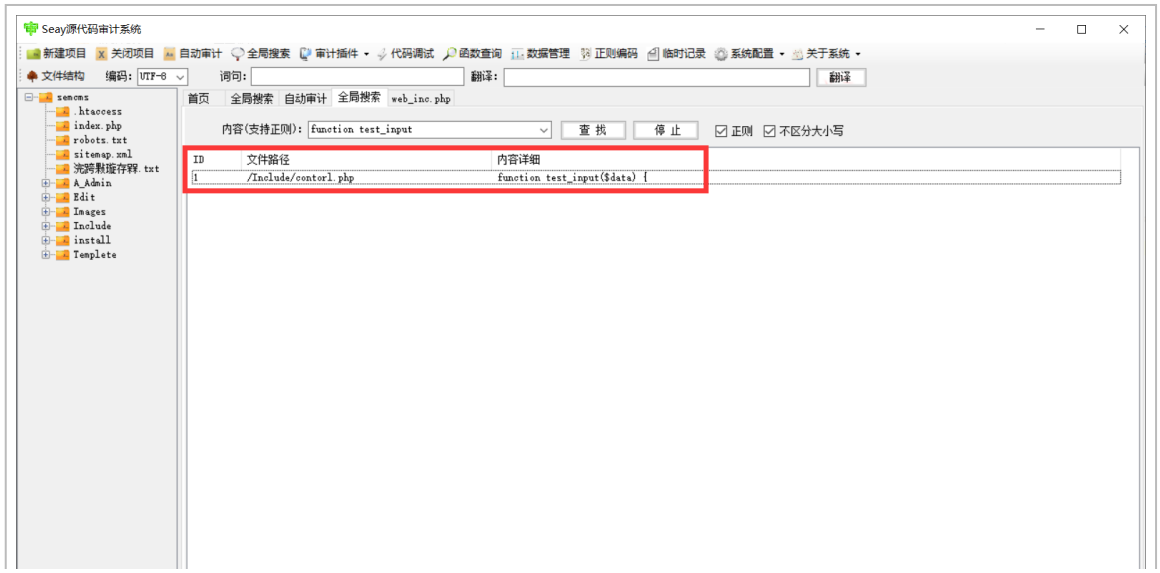
看一下 $language 参数传参是从哪个点进来的，这样方便我们在网页中找到注入点

通过查询发现，$language 是在 `web_inc.php` 文件中，`$Language=test_input(verify_str($_POST["languageID"]` ，从这里可以看到 $languageID 要经过两层函数的过滤才能将值赋给 $language，一层是我们刚刚看到的 `verify_str（）函数` ，通过调用 sql 过滤函数来过滤 sql 语句，于是我们分析 `test_input()函数`



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-fad289a49034beb3dee9651f1ec21a84fc53d50c.png)

通过全局搜索发现 `test_input（）函数` 构建在 `/Include/contorl.php` 文件中

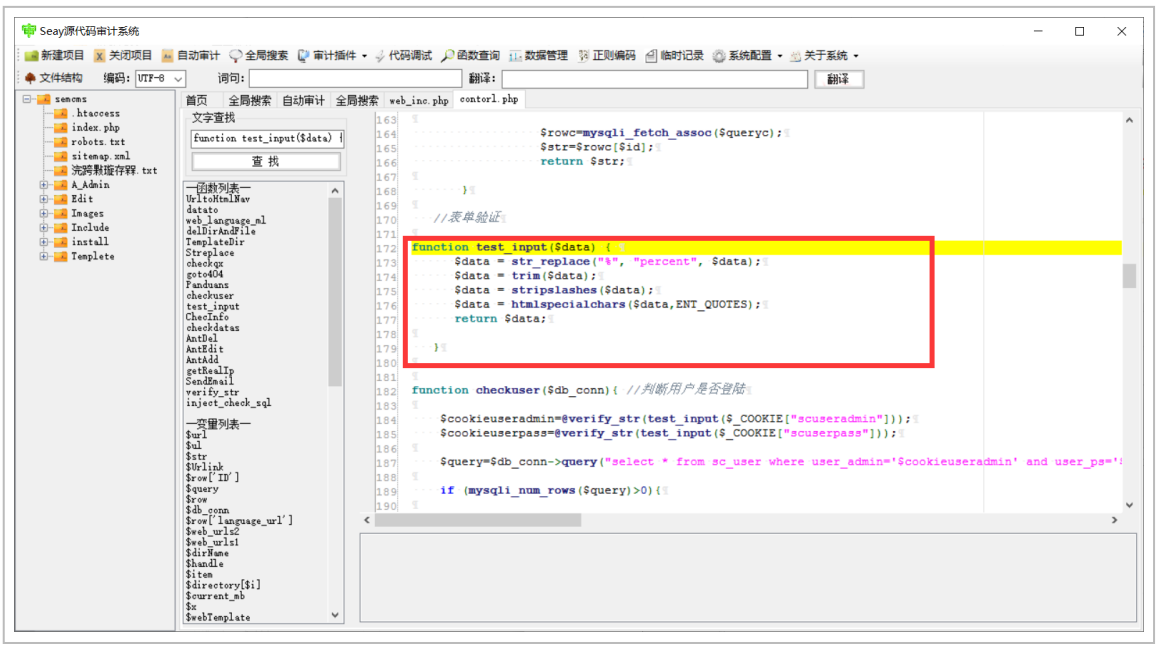


(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-84bf36b1b9dc2a2b541d506581d7bf6d74fb735c.png)

test_input（）函数分析

---

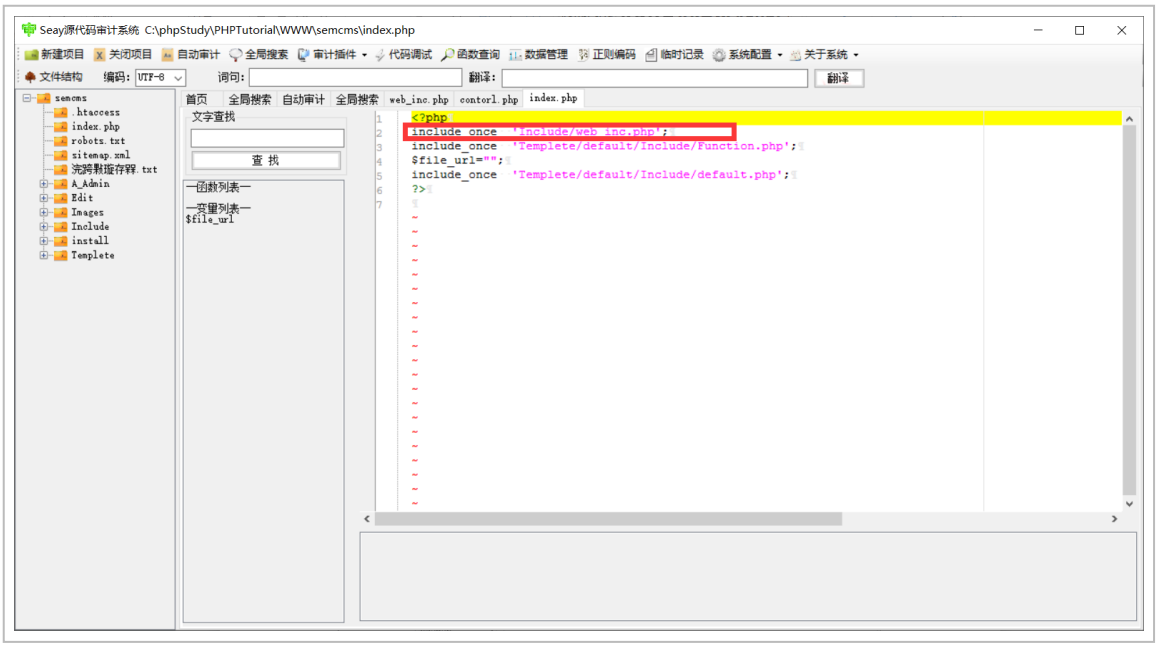
1. `str_replace()函数` 将传进来的 $data 参数进行匹配，如果参数中的字符串含有 % 的话，将其替换为 percent
2. `trim（）函数` 用于去除 $data 字符串首尾空格、解转义、实体编码、NULL 这些字符串
3. `stripslashes（）函数` 用于删除反斜杠
4. `htmlspecialchars（）函数` 用于转换为 html 实体化, ENT_QUOTES 参数用于处理单引号和双引号为 html 实体化



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-79e232d71c59745063f46b972d79a0fcc13fd2fb.png)

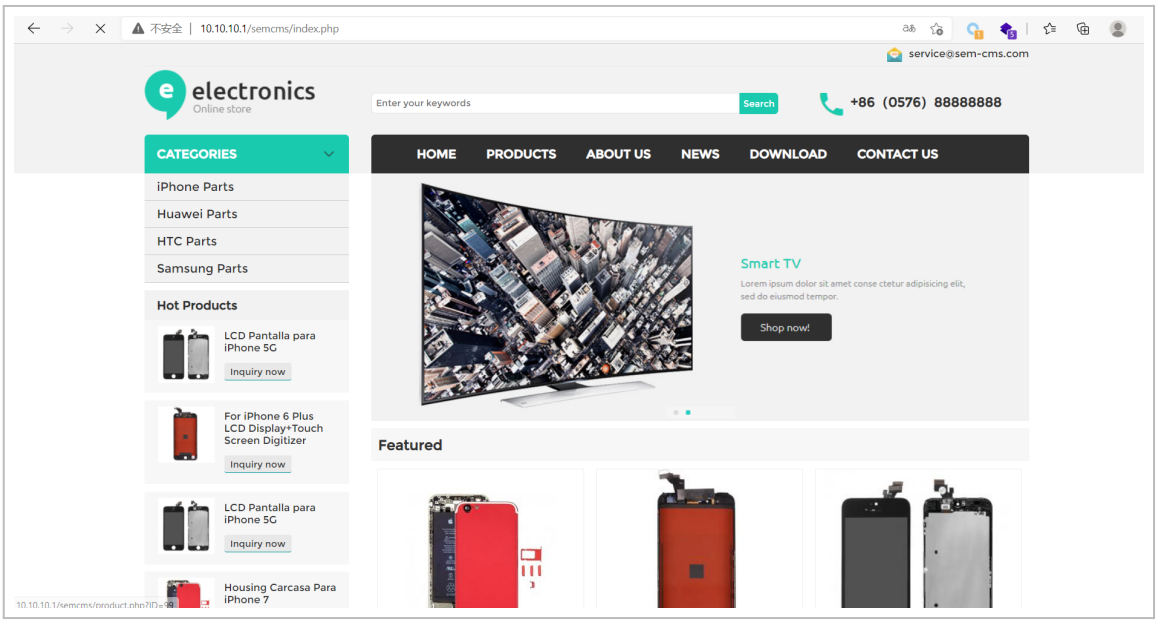
三、SQL 漏洞利用

`web_inc.php` 是一个配置文件，在其他的文件中也是都包含这个文件的



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-8c9d9b8fed5d9588b22459b518ed3af91a3a0db1.png)

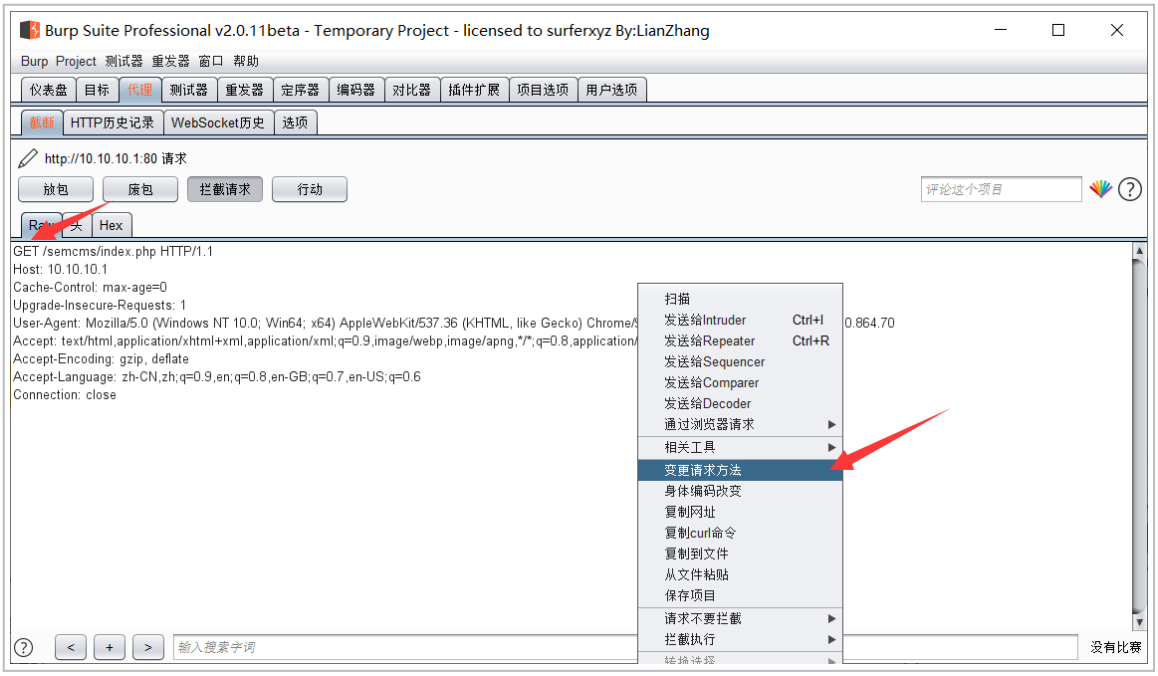
所以我们就直接去 index.php 去尝试构造 payload 看能不能利用成功。



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-

9dd94e6d177d9deb92a3e6d6513e11ae43a5d896.png)

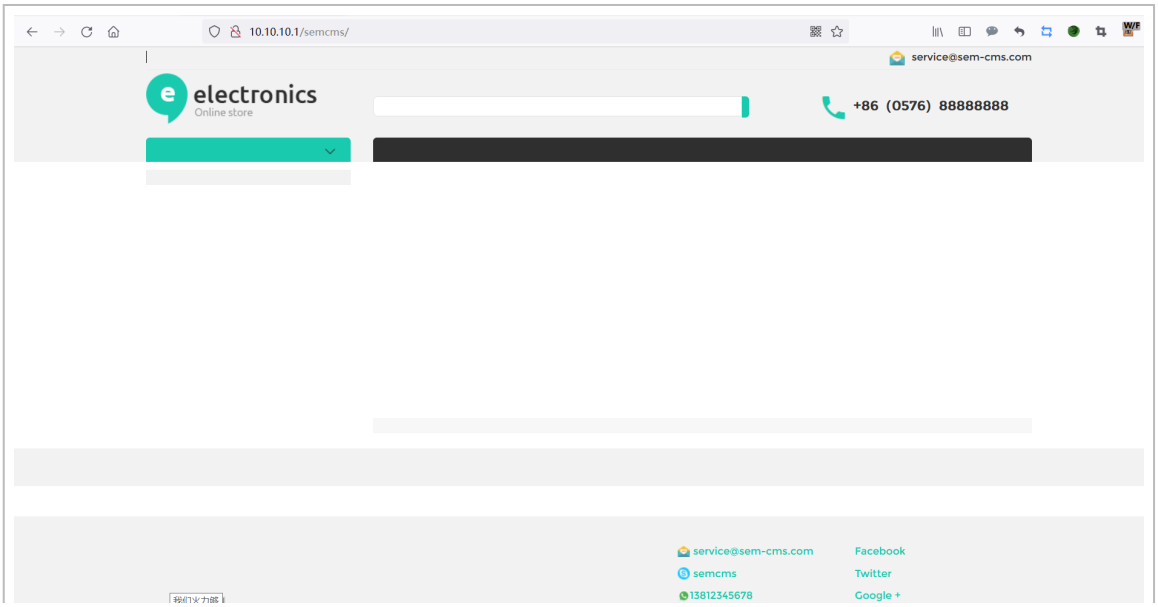
通过 burpsuite 软件抓包，获取网页请求包，然后将请求方法变更为 `POST传参` ，这时候可以在末尾传入我们的参数 `languageID`



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-770392fed225d960ada2aaa42d53d37204ef4aa4.png)

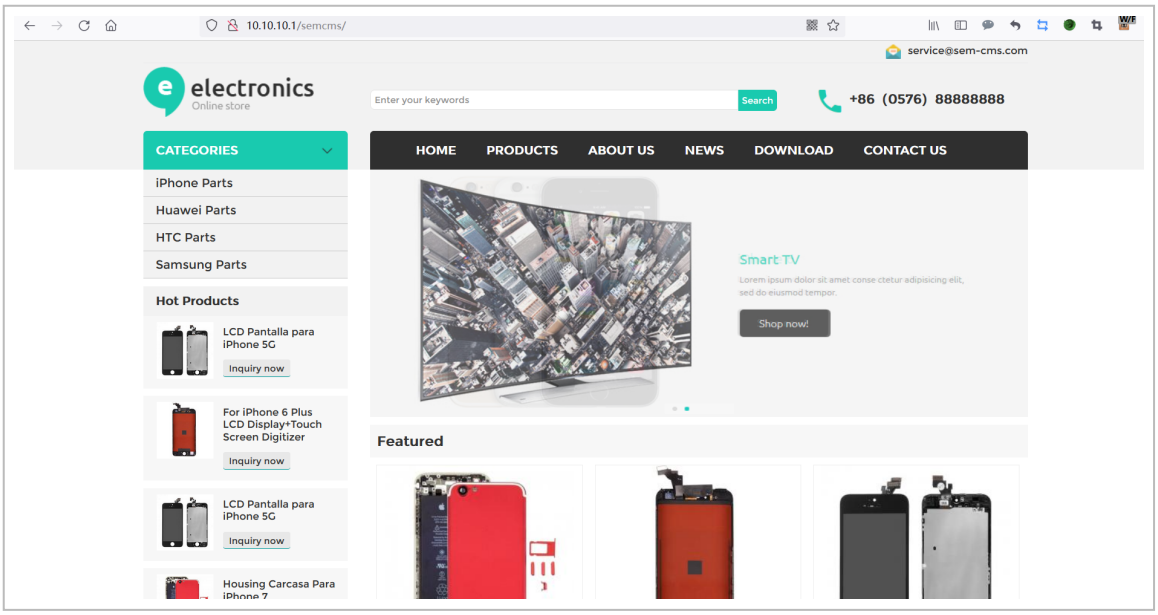
因为 `verify_str（）函数` 调用的 SQL 过滤语句中没有过滤掉 and、sleep() 函数，所以可以构造 SQL 延时注入，也可以构造布尔型盲注

当参数为 languageID=1 and 3>100 时，由于逻辑错误，返回的页面错误



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-0650858f1d26db18d2561f632b0ed339c481b6ae.png)

当参数为 languageID=1 and 3>2 时，由于逻辑正确，返回的页面正确

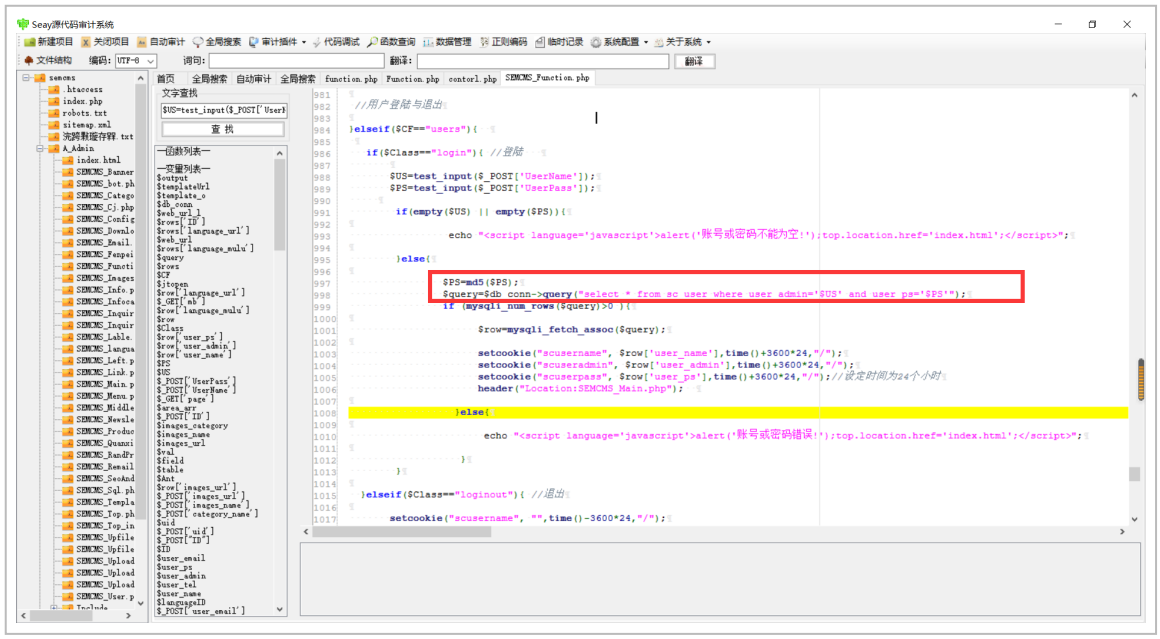


(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-2cfafa619183f8648c0c935a849162eb8e19feda.png)

四、暴力破解漏洞

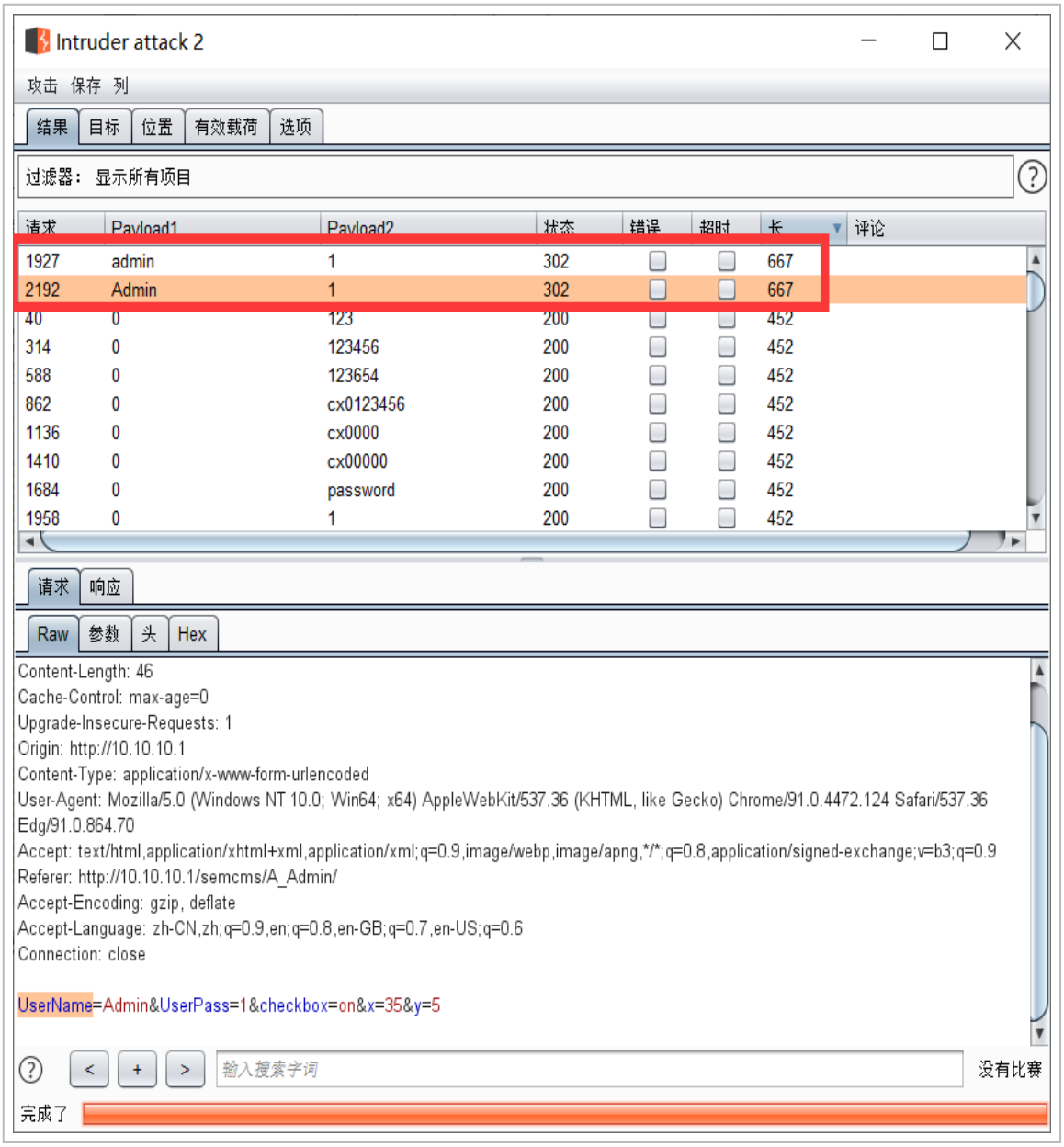
在 semcms/A_Admin/SEMCMS_Function.php 文件中，有一段验证前端账号密码登录的代码

1. 通过 POST 传参将账号密码传入
2. 如果账号或者密码为空则弹窗“账号密码不能为空”
3. 将传入的密码通过 $md5 函数加密，然后再进行 sql 语句的匹配查询，所以不能利用万能密码登录后台，因为密码会被 md5 进行一次加密
4. 如果查询成功，则为用户添加 cookie



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-341b16d2f5a05a90c87c57762010452c6e715073.png)

虽然不能用万能密码绕过数据库验证，但是该登录界面没有登录此数限制也没有验证码校验，所以尝试利用暴力破解登录后台，最后得到的账号密码为 `Admin/1`



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-898d1243ca0031a4afbfc54db9fe131a51a32855.png)

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-d5800de90cb70534bb15e80c7e21a8495b99d447.png)

五、信息泄露漏洞

在刚刚的前端登录验证的代码总，当用户登录成功的时候会给用户发送 setcookie 信息，其中 `setcookie（）函数` 构造是这样的，设置三个参数 `scusername、scuseradmin、scuserpass`，分别将 user_name,user_admin,user_ps 的值赋给它们，从而造成了信息泄露，如下所示

---

scusername=%E6%80%BB%E8%B4%A6%E5%8F%B7;
scuseradmin=Admin;
scuserpass=c4ca4238a0b923820dcc509a6f75849b

```
}elseif($CF=="users"){
  if($Class=="login"){ //登陆
      $US=test_input($_POST['UserName']);
      $PS=test_input($_POST['UserPass']);
       if(empty($US) || empty($PS)){
                echo "<script language='javascript'>alert('账号或密码不能为空!');top.location.href='index.html';</script>";
       }else{
```

```
            $PS=md5($PS);
            $query=$db_conn->query("select * from sc_user where user_admin='$US' and user_ps='$PS'");
            if (mysqli_num_rows($query)>0 ){

                $row=mysqli_fetch_assoc($query);

                setcookie("scusername", $row['user_name'],time()+3600*24,"/");
                setcookie("scuseradmin", $row['user_admin'],time()+3600*24,"/");
                setcookie("scuserpass", $row['user_ps'],time()+3600*24,"/");//设定时间为24个小时
                header("Location:SEMCMS_Main.php");

            }else{

                echo "<script language='javascript'>alert('账号或密码错误!');top.location.href='index.html';</script>";

            }
        }
```

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-bc5aa854d371907b8fe1e150082592aac5b2e35a.png)

将 `scuserpass=c4ca4238a0b923820dcc509a6f75849b` 放到 md5 解密得到密码

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-0d6582a59e6001c9eab5e545fc702cad919ea94f.png)

总结

用 Seay 源代码审计系统扫描出来的还有 XSS 注入漏洞，还有文件包含漏洞，文件包含漏洞用函数 include_once 调用文件，但是没有用参数来获取这个调用，可能不存在文件包含漏洞。xss 注入漏洞因为过滤了 < 符号，在构造的时候可能会被过滤掉。