ctf 选择题 题库_NISA CTF Web部分题库WriteUp

原创

weixin_39779032 于 2020-12-19 21:44:41 发布 209 收藏
文章标签： ctf选择题 题库

本文链接：https://blog.csdn.net/weixin_39779032/article/details/111542677

第一题 Just_post

点进去直接显示源码了

```
$flag=$_POST["flag"];

if($flag=='NISA_CTF')

echo 'NISA_CTF{******}';
```

根据这个源码，直接给服务器POST flag=NISA_CTF，就直接返回flag了

第二题 easy

点进去还是直接源码<?php

```
include "flag.php";

$hello = @$_GET['hello'];

$id = @$_GET['id'];

if(md5($hello) == 0 && !is_numeric($id) && $id == 1){

echo "$flag";

}

show_source(__FILE__);

?>
```

看到这里对传入的id有一个判断，不过这个判断很奇怪，用is_numeric($id)来过滤数字，但是又要求id=1，这就自相矛盾了。

这时候就要用到php神奇的比较符了，这里分享一下我的学习笔记：

.php中有两种比较的符号 == 和 === 。=== 在进行比较的时候，会先判断两种字符串的类型是否相等，再比较；== 在进行比较的时候，会先将字符串类型转化成相同，再比较。

如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换成数值，并且比较按照数值来进行。

如果是比较数字和数字加字符的时候，==会比较数字和字符串开头的数字(直到出现非数字的部分)

好了，这题的解法就显而易见了，我们只需要传入id=1a (只要是1后面跟上任何的数字都行)就可以获得flag了

第三题 easy_py

这题挺智障的。。。直接观察url，返回上一级菜单，就会发现每个子网页文件都存了一个字母，连在一起就是flag了。。会写py的就写个脚本跑下来吧，不过这个flag也不长全部手工也行。

第四题 简单绕过

出来的是源码：

```
$temp = $_GET["password"];

is_numeric($temp)?die("no numeric"):NULL;

if($temp>76461924)

{

echo $flag;

}
```

发现这题和第二题有点像，不过这次在判断是否是数字以后还要大于某个数字。这时候就要用到我们is_numberic()函数的一个漏洞了，这里再分享一下我的学习笔记：

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于第一个空格字符会跳过空格字符判断，接着后面的判断。

所以我们只需要传入一个比76461924大的数字，并且在后面加上一个%00(空字符)，就可以绕过is_numberic()函数的检查啦~

第五题 律师函警告

点击View-source会显示出源码:

```
require("flag.php");

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

highlight_file(__FILE__);

die();

}

if (isset($_GET['amazing_word'])) {$what_he_said = $_GET['amazing_word'];

$what_you_dont_want_to_hear = 'you_ctf_like_cxk';

$what_you_actually_heard = preg_replace("/$what_you_dont_want_to_hear/", 'c', $what_he_said);

if ($what_you_actually_heard === $what_you_dont_want_to_hear) {

show_me_flag();

}

}

?>
```

这题的关键部分就是
$what_you_actually_heard = preg_replace("/$what_you_dont_want_to_hear/", 'c', $what_he_said);在题目的最后会验证$what_you_actually_heard === $what_you_dont_want_to_hear(也就是要求$what_you_actually_heard==you_ctf_like_cxk 但是在前面的preg_replace的过程中会把$what_he_said中的you_ctf_like_cxk部分替换成c 也就是说如果传入一个you_ctf_like_cxk，最终得到的是一个c。

这时候我们就要用到一个preg_replace()函数的bug(特性)了：

preg_replace()函数在完成一次替换工作以后不会从头开始扫描替换项而是从替换后的位置开始扫描

所以我们只需要利用这个特性，把you_ctf_like_cxk中的c提前换为you_ctf_like_cxk变成you_you_ctf_like_cxktf_like_cxk再传入就可以得到flag了(不懂得话可以在草稿纸上手动替换下试试看)

第六题 饼干秘方

这题题目已经提示饼干(cookie)了，那我们直接查看下cookie看看：

发现cookie里面已经有了username和passwd两项，而且有效期都是10秒，查看这两个数据的值，发现什么是一堆的乱码。这时候如果对密码学比较敏感的同学应该能发现后面几位看起来好像有点熟悉的感觉...没错！就是md5！解开来之后发现用户名对应的是guest，密码对应的是tseug

对网站攻击实战比较熟悉的同学可能会留意到常用的网站登陆账号除了guest还有一个很重要的admin，所以我们反其道行之，用admin和nimda(和前面解出来的guest一样的套路)重新加密回去，得到这两个的md5，在像刚刚一样包装回cookie(注意下位数要一致，也就是要在md5前面补6个字符)，抓包，改包，既可以拿到flag啦~~~

这题脑洞实在有点大....最后一步我也是和出题人交流之后才反应过来的....(/挠头)