攻防世界web进阶区web2详解

原创

無名之涟 于 2020-07-31 12:04:48 发布

分类专栏： CTF

本文链接：https://blog.csdn.net/hxhxhxhxx/article/details/107710072

攻防世界web进阶区web2详解

题目

```php
<?php
$miwen="a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws";

function  encode($str){
        $_o=strrev($str);
        //  echo  $_o;

        for($_0=0;$_0<strlen($_o);$_0++){

                $_c=substr($_o,$_0,1);
                $__=ord($_c)+1;
                $_c=chr($__);
                $_=$_.$_c;
        }
        return  str_rot13(strrev(base64_encode($_)));
}

highlight_file(__FILE__);
/*
      逆向加密算法，解密$miwen就是flag
*/
?>
```


解法

```php
<?php
$miwen="a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws";

function encode($str){
  $_o=strrev($str);
  // echo $_o;

  for($_0=0;$_0<strlen($_o);$_0++){

    $_c=substr($_o,$_0,1);
    $__=ord($_c)+1;
    $_c=chr($__);
    $_=$_.$_c;
  }
  return str_rot13(strrev(base64_encode($_)));
}

highlight_file(__FILE__);
/*
  逆向加密算法，解密$miwen就是flag
*/
?>
```

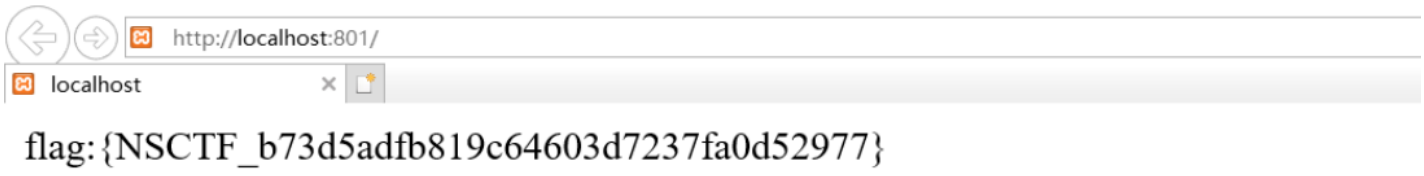
我们这里来审计一波

for循环，从0到循环到字符串长度

从开始之后_0长度的字符开始寻找，长度为1的输出给了$_c

php脚本如下

```php
<?php
$miwen="a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws";
$miwen=base64_decode(strrev(str_rot13($miwen)));

//echo $miwen;

$m=$miwen;

for($i=0;$i<strlen($m);$i++){

 $_c=substr($m,$i,1);
 $__=ord($_c)-1;   # 字符转数字，在减1
 $__=chr($__);    # 数字转字符

   $_=$_.$__;       # 拼接字符串

}

echo strrev($_);      # 反转字符串
```



flag:{NSCTF_b73d5adfb819c64603d7237fa0d52977}

strrev函数

反转字符串

```php
<!DOCTYPE html>
<html>
<body>

<?php
echo strrev("I love Shanghai!");
?>

</body>
</html>
```

```
!iahgnahS evol I
```


substr

源代码:

```
<!DOCTYPE html>
<html>
<body>

<?php
echo substr("Hello world",6);
?>

</body>
</html>
```

运行结果

world

从第六位开始，返回之后的值

定义和用法

substr() 函数返回字符串的一部分。

注释：如果 start 参数是负数且 length 小于或等于 start，则 length 为 0。

语法

```
substr(string,start,length)
```

参数	描述
string	必需。规定要返回其中一部分的字符串。
start	必需。规定在字符串的何处开始。 • 正数 - 在字符串的指定位置开始 • 负数 - 在从字符串结尾的指定位置开始 • 0 - 在字符串中的第一个字符处开始
length	可选。规定要返回的字符串长度。默认是直到字符串的结尾。 • 正数 - 从 start 参数所在的位置返回 • 负数 - 从字符串末端返回

ord

源代码:

```
<!DOCTYPE html>
<html>
<body>

<?php
echo ord("h")."<br>";
echo ord("hello")."<br>";
?>

</body>
</html>
```

运行结果

104
104

返回第一个字母的ASCII

定义和用法

ord（）函数返回字符串中第一个字符的ASCII值。

语法

```
ord(string)
```

参数	描述
string	必需。要从中获得 ASCII 值的字符串。

技术细节

返回值:	以整体形式返回ASCII值。
PHP版本:	4+

chr()

定义和用法

chr（）函数从指定ASCII值返回字符。

ASCII值可被指定为十进制值，八进制值或十六进制值。八进制值被定义为带前缀0，十六进制值被定义为带前缀0x。

语法

```
chr(ascii)
```

参数	描述
ascii	必需。ASCII 值。

技术细节

返回值:	返回指定的字符。
PHP版本:	4+

源代码:

```
<!DOCTYPE html>
<html>
<body>

<?php
echo chr(52) . "<br>"; // Decimal value
echo chr(052) . "<br>"; // Octal value
echo chr(0x52) . "<br>"; // Hex value
?>

</body>
</html>
```

运行结果

4
*
R

从指定的ascii值返回字符
4的十进制ascii是52

str_rot13()

PHP str_rot13() 函数

PHP String 参考手册

实例

编码并解码字符串：

```
<?php
echo str_rot13("Hello World");
echo "<br>";
echo str_rot13("Uryyb Jbeyq");
?>
```

运行实例 »

定义和用法

str_rot13() 函数对字符串执行 ROT13 编码。

ROT13 编码是把每一个字母在字母表中向前移动 13 个字母得到。数字和非字母字符保持不变。

提示：编码和解码都是由相同的函数完成的。如果您把一个已编码的字符串作为参数，那么将返回原始字符串。

语法

```
str_rot13(string)
```



```
<!DOCTYPE html>
<html>
<body>

<?php
echo str_rot13("Hello World");
echo "<br>";
echo str_rot13("Uryyb Jbeyq");
?>

</body>
</html>
```

Uryyb Jbeyq
Hello World

一种编码解码函数

对字符串执行 ROT13 转换,ROT13 编码简单地使用字母表中后面第 13 个字母替换当前字母，同时忽略非字母表中的字符。编码和解码都使用相同的函数，传递一个编码过的字符串作为参数，将得到原始字符串。