hackergame misc writeup

原创

~VAS~ 于 2022-01-11 11:13:00 发布 412 收藏

分类专栏： 笔记 ctf 文章标签： 网络安全

本文链接：https://blog.csdn.net/zip471642048/article/details/122427440

版权

笔记 同时被 2 个专栏收录

53 篇文章 0 订阅

订阅专栏

ctf

50 篇文章 1 订阅

订阅专栏

猫咪电路

mc的红石电路,逆推过程,输入的二进制01就是flag

flag{0110101000111100101111111111111111111010}

猫咪和键盘

纵向随机切割

```
with open("typed_printf.cpp","r") as f:
    lines=f.readlines()
    #print(lines)
    for line in lines:
        seg1=line[0:1]
        seg2=line[1:7]
        seg3=line[8:20]
        seg4=line[20:22]
        seg5=line[22:32]
        seg6=line[32:39]
        seg7=line[39:-1]
        print((seg1+seg6+seg2+seg4+seg3+seg5+seg7).strip())
```

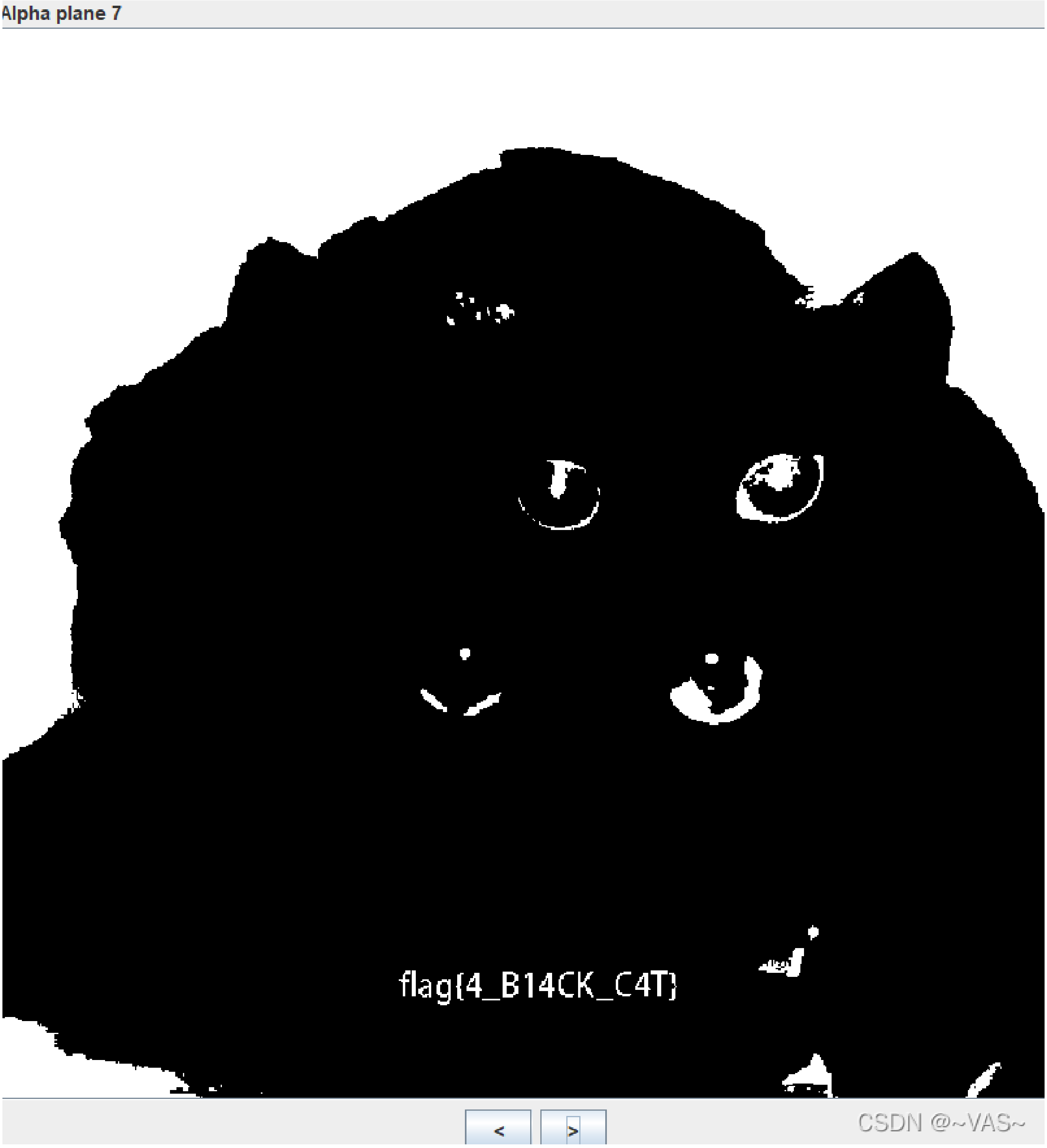
```
/*
* name: typed_printf.cpp
* compile: g++ -std=c++17 typed_printf.cpp
* title: type safe printf
* author: nicekingwei
* url: aHR0cHM6Ly96anUtbGFtYmRhLnRlY2gvY3BwZHQtcHJpbnRmLw==
* related knowledge:
*   - value and type
*       value->value: function
*       type->value: parametric polymorphism
*       type->type: generic
*       value->type: dependent type
*  - auto
*  - if constexpr
*/
#include <iostream>
#include <functional>
#include <type_traits>

using namespace std;

template<const char*format>
static auto println() {
if constexpr (format[0]=='%') {
if constexpr (format[1]=='d') {
return [](int x){cout<<x<<endl;};
} else if constexpr (format[1]=='s') {
return [](const char* x){cout<<x<<endl;};
} else {
return "error";
}
} else {
return "error";
}
}

struct unit_t {char x;};

template<typename T,typename R>
constexpr auto get_arg(R (*f)(T)){
return T{};
}

template<typename T>
constexpr bool cont_takes_no_arg(T cont){
using cont_t = decay_t<T>;
using arg_type = decay_t<decltype(get_arg(cont))>;
return is_same<unit_t,arg_type>::value;
}


template<typename T,typename R,typename X,R (*cont)(X)>
auto print_var(T x){
cout<<x;
return cont;
}

template<typename T,typename R,typename X,R (*cont)(void)>
auto print_var(T x){
cout<<x;
return cont();
}

template<char c,typename R,typename X,R (*cont)(X)>
auto print_const(X x){
cout<<c;
return cont(x);
}

template<char c,typename R,typename X,R (*cont)(void)>
auto print_const(){
```

```
cout<<c;
return cont();
}


template<typename R,typename X>
constexpr auto cont_ret_type(R (*cont)(X)){
return R{};
}

template<typename R>
constexpr auto cont_ret_type(R (*cont)()){
return R{};
}

template<typename R,typename X>
constexpr auto cont_arg_type(R (*cont)(X)){
return X{};
}

template<typename R>
constexpr auto cont_arg_type(R (*cont)()){
return unit_t{};
}

unit_t print_nothing(){return unit_t{};}

#define cont_ret_t decay_t<decltype(cont_ret_type(cont))>
#define cont_arg_t decay_t<decltype(cont_arg_type(cont))>

template<const char*format,int i>
constexpr auto _typed_printf(){
if constexpr (format[i]=='%' && format[i+1] == 'd') {
constexpr auto cont = _typed_printf<format,i+2>();
return print_var<int,cont_ret_t,cont_arg_t,cont>;
} else if constexpr (format[i]=='%' && format[i+1] == 's') {
constexpr auto cont = _typed_printf<format,i+2>();
return print_var<const char*,cont_ret_t,cont_arg_t,cont>;
} else if constexpr (format[i]!='\0') {
constexpr auto cont = _typed_printf<format,i+1>();
return print_const<format[i],cont_ret_t,cont_arg_t,cont>;
} else {
return print_nothing;
}
}

#define def_typed_printf(f,str) constexpr static const char str_fmt##f[] = str; auto f = _typed_printf<str_fmt##
f,0>();

#define ABC "FfQ47if9Zxw9jXE68VtGA"
#define BAC "JDk6Y6Xc88UrUtpK3iF8p"
#define CAB "7BMs4y2gzdG8Ao2gv6aiJ"

int main(){
def_typed_printf(f_l_x_g_1, "%s%s%s%s");
f_l_x_g_1("fl")("a")("g")("{");
def_typed_printf(a_a_a_a_a_a_a_a_a, "%s%s%s%s%s%s%d");
a_a_a_a_a_a_a_a_a(ABC)("")(BAC)("")(CAB)("")('}');
def_typed_printf(def_typed_printf_, "%s%d%s");
def_typed_printf_("typed_printf")('_')("}");
```

```
return 0;
}
```

c++17运行得flag

白与夜

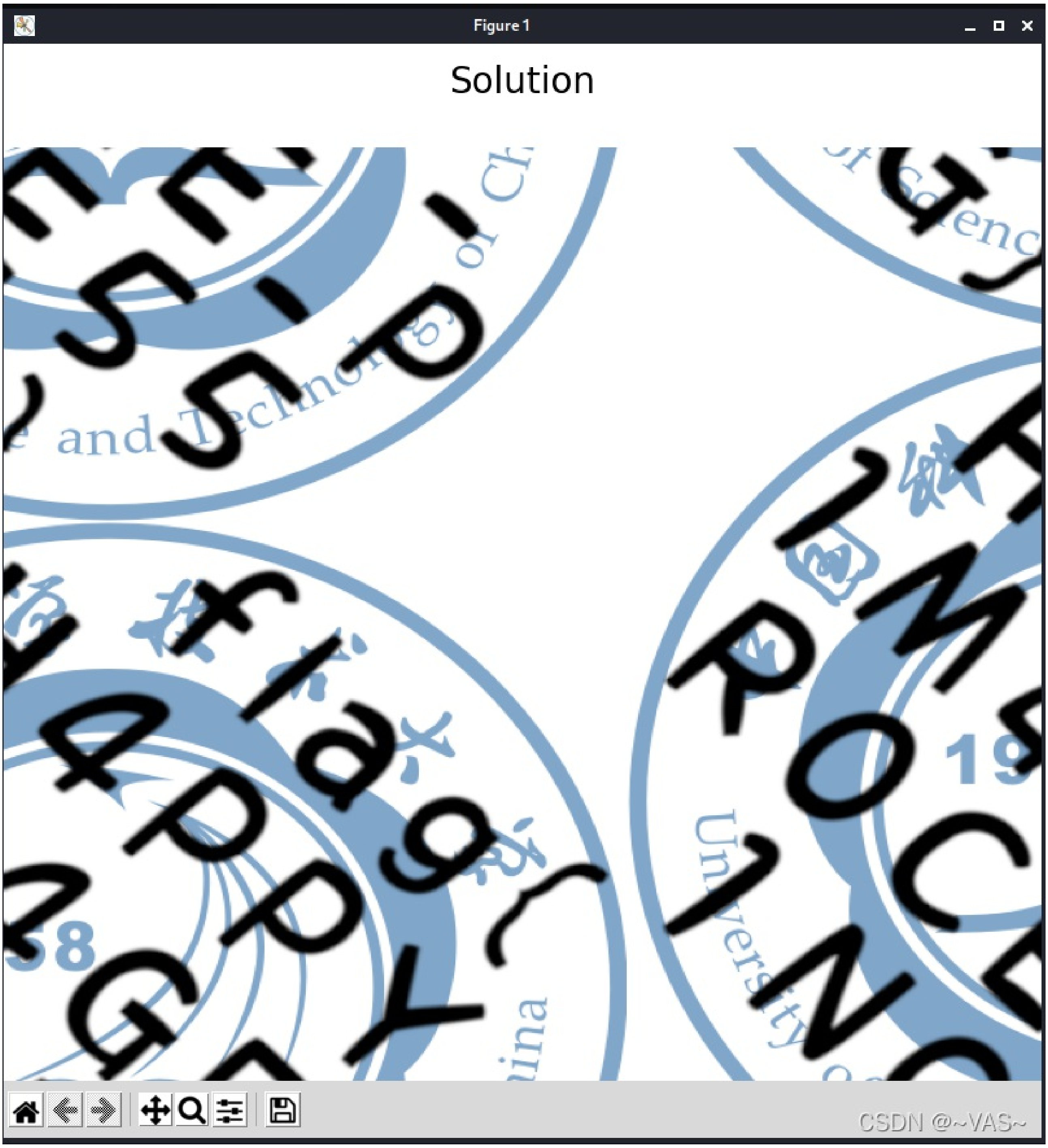


lsb隐写

游园会的集章卡片

用 montage和gaps拼图,我不会用这个东西,拼歪了只能硬着把flag弄出来了

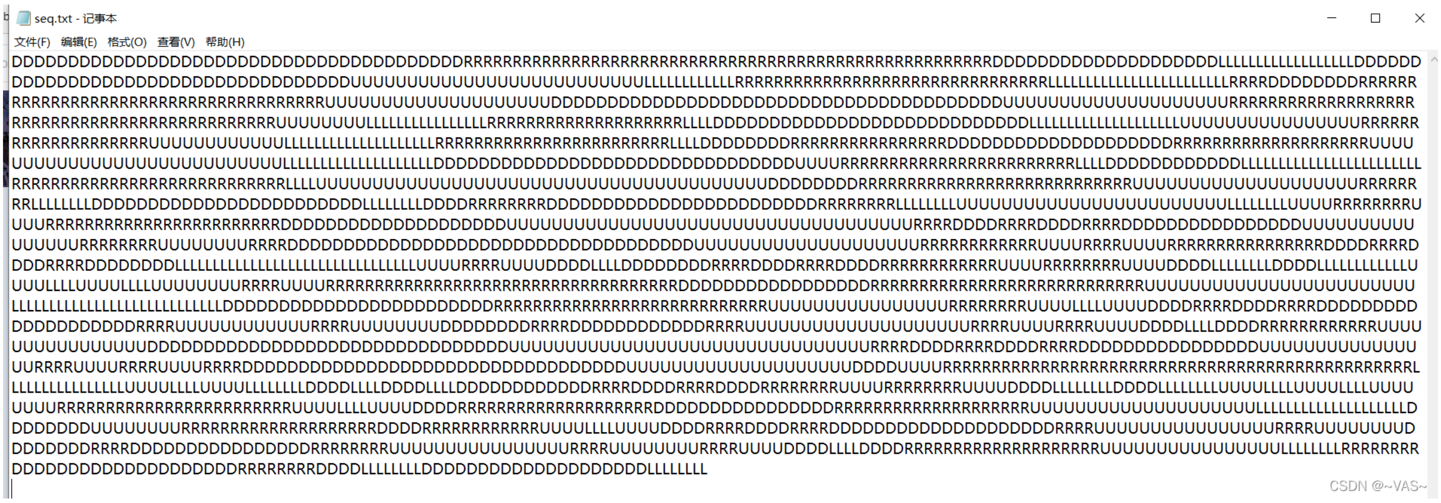
```
montage *png -tile 5x5 -geometry 125x125+0+0 flag.png
gaps --image=flag.png --generations=50 --population=25 --size=125
```



flag{H4PPY_1M4GE_PR0CE551NG}

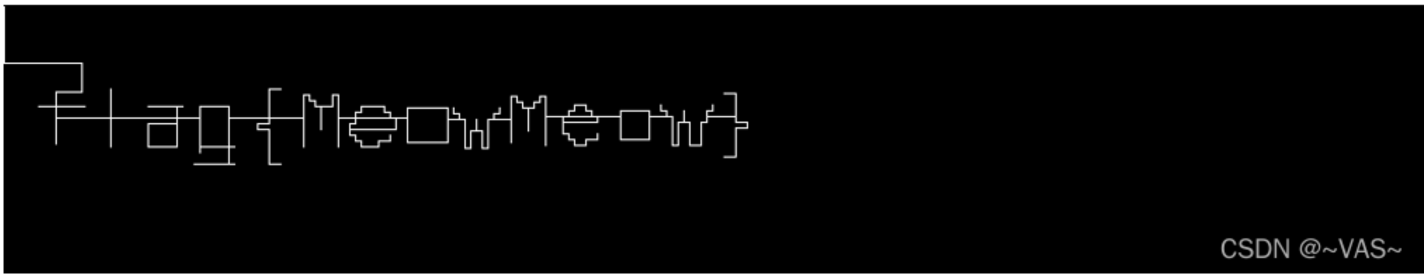
猫咪遥控器

把txt内容处理成图像



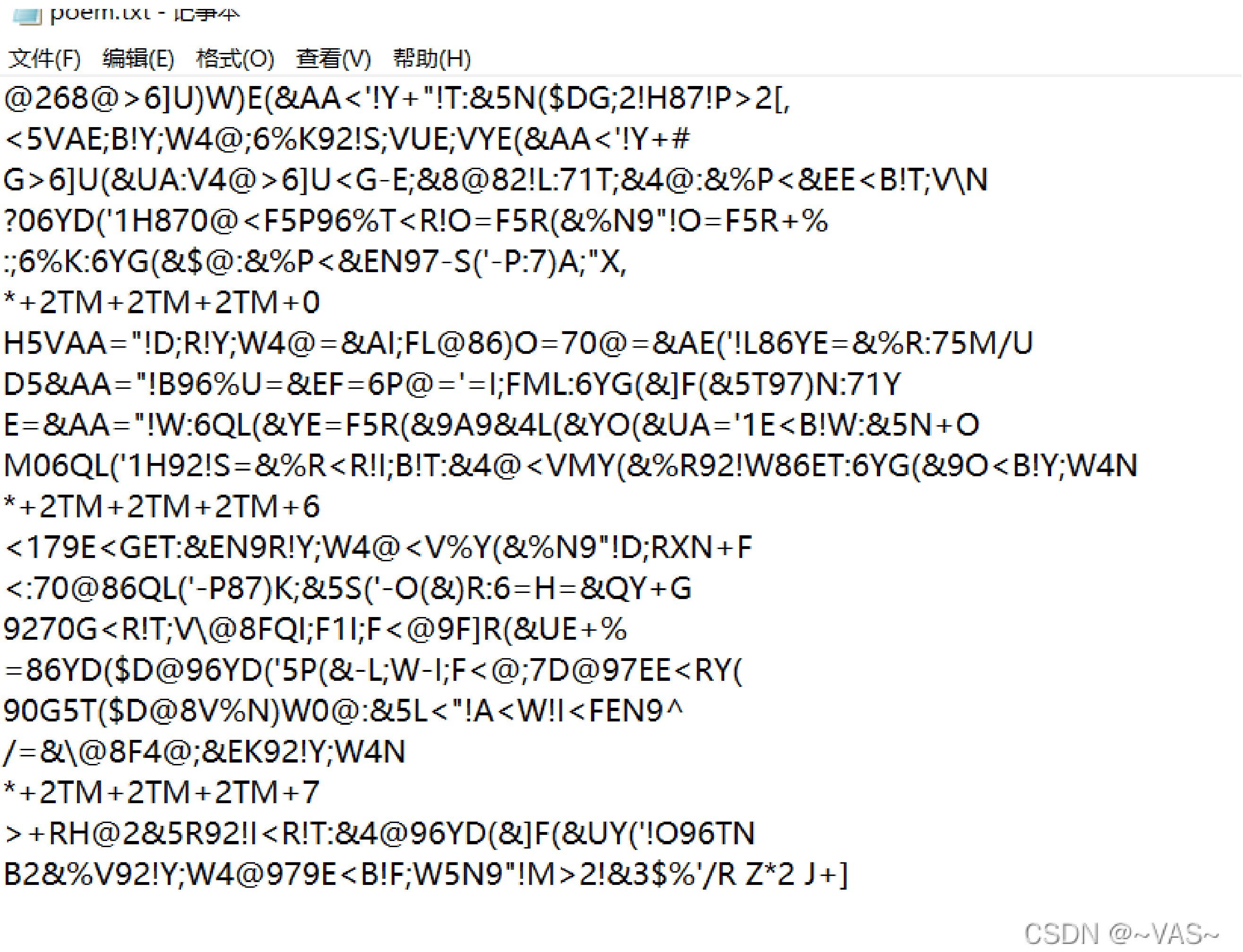
```
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRDDDDDDDDDDDDDDDDDDDDDDDLLLLLLLLLLLLLLLLLLLDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUUUUUUUUUUUUUUUULLLLLLLLLLLRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRLLLLLLLLLLLLLLLLLLLLLLLLLRRRRDDDDDDDDDRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRUUUUUUUUUUUUUUUUUUUUUUUDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUUUUUUUURRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRUUUUUUUULLLLLLLLLLLLLLLLRRRRRRRRRRRRRRRRRRRLLLLDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDLLLLLLLLLLLLLLLLLLUUUUUUUUUUUUUUUURRRRRR
RRRRRRRRRRRRRUUUUUUUUUUUULLLLLLLLLLLLLLLLLLLLRRRRRRRRRRRRRRRRRRRRRRRRRLLLLDDDDDDDDRRRRRRRRRRRRRRRRRDDDDDDDDDDDDDDDDDDDDDRRRRRRRRRRRRRRRRRRRRRUUUU
UUUUUUUUUUUUUUUUUUUUUUUUULLLLLLLLLLLLLLLLLLLLDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDUUURRRRRRRRRRRRRRRRRRRRRRRRLLLLDDDDDDDDDDDDLLLLLLLLLLLLLLLLLLLLLLL
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRLLLLUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUDDDDDDDDRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRUUUUUUUUUUUUUUUUUUUUURRRRRR
RRLLLLLLLLDDDDDDDDDDDDDDDDDDDDDDDDDDLLLLLLLLDDDDRRRRRRRRDDDDDDDDDDDDDDDDDDDDDDDDDDDRRRRRRRRLLLLLLLLUUUUUUUUUUUUUUUUUUUUUUUUULLLLLLLLUUUURRRRRRRRU
UUURRRRRRRRRRRRRRRRRRRRRRRRRRRRDDDDDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUURRRRDDDDRRRRDDDDRRRRDDDDDDDDDDDDDDDDDDDUUUUUUUUUUU
UUUUUURRRRRRRUUUUUUUURRRRDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUUUUUUUURRRRRRRRRRRRUUUURRRRUUUURRRRRRRRRRRRRRRRRRRDDDDRRRRD
DDDRRRRDDDDDDDDLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLUUUURRRRUUUUDDDDLLLLDDDDDDDDDRRRRDDDDRRRRDDDDRRRRRRRRRRUUUURRRRRRRRUUUUDDDDLLLLLLLLDDDDLLLLLLLLLLLLLU
UUULLLUUUULLLLUUUUUUUURRRRUUUURRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRDDDDDDDDDDDDDDDDDDDDRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRUUUUUUUUUUUUUUUUUUUUUUUU
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLDDDDDDDDDDDDDDDDDDDDDDDDDDDRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRUUUUUUUUUUUUUUUUURRRRRRRRUUUULLLLUUUUDDDDRRRRDDDDRRRRDDDDDDDDD
DDDDDDDDDDDRRRRUUUUUUUUUUUUURRRRUUUUUUUUDDDDDDDDRRRRDDDDDDDDDDDDDRRRRUUUUUUUUUUUUUUUUUUUUUURRRRUUUURRRRUUUUDDDDLLLLDDDDRRRRRRRRRRRRRRUUUU
UUUUUUUUUUUUUDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUURRRRDDDDRRRRDDDDRRRRDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUU
UURRRRUUUURRRRUUUURRRRDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUUUUUUUDDDDUUUURRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRL
LLLLLLLLLLLLLLLUUUULLLLUUUULLLLLLLLDDDDLLLLDDDDLLLLDDDDDDDDDDDDRRRRDDDDRRRRDDDDRRRRRRRRRUUUURRRRRRRRRUUUUDDDDLLLLLLLLDDDDLLLLLLLLUUUULLLLUUUULLLLUUUU
UUUURRRRRRRRRRRRRRRRRRRRRRRRRRRRUUUULLLLUUUUDDDDRRRRRRRRRRRRRRRRRRRRRDDDDDDDDDDDDDDDDDDRRRRRRRRRRRRRRRRRRRRRUUUUUUUUUUUUUUUUUUUUULLLLLLLLLLLLLLLLLLLD
DDDDDDDUUUUUUUURRRRRRRRRRRRRRRRRRRRRRDDDDRRRRRRRRRRRRRUUUULLLLUUUUDDDDRRRRDDDDRRRRDDDDDDDDDDDDDDDDDDDDRRRRUUUUUUUUUUUUUUUUURRRRUUUUUUUUD
DDDDDDDRRRRDDDDDDDDDDDDDDDDDDRRRRRRRRUUUUUUUUUUUUUUUURRRRUUUUUUUURRRRUUUUDDDDLLLLDDDDRRRRRRRRRRRRRRRRRRRRRRRUUUUUUUUUUUUUUUUUUULLLLLLLLRRRRRRRR
DDDDDDDDDDDDDDDDDDDDDDRRRRRRRRDDDDLLLLLLLLDDDDDDDDDDDDDDDDDDDDDDDDLLLLLLLL
```

```
from PIL import  Image

image= Image.new("RGB",(1000,1000))

f= open('seq.txt','r').read()
x=0
y=0

for i in f:
    if(i=='D'):
        y+=1
    elif(i=='L'):
        x-=1
    elif(i=='R'):
        x+=1
    elif(i=="U"):
        y-=1
    print(x,y)
    image.putpixel((x,y),(255,255,255))
image.show()
```



她的诗

uuencode隐写类似base64隐写



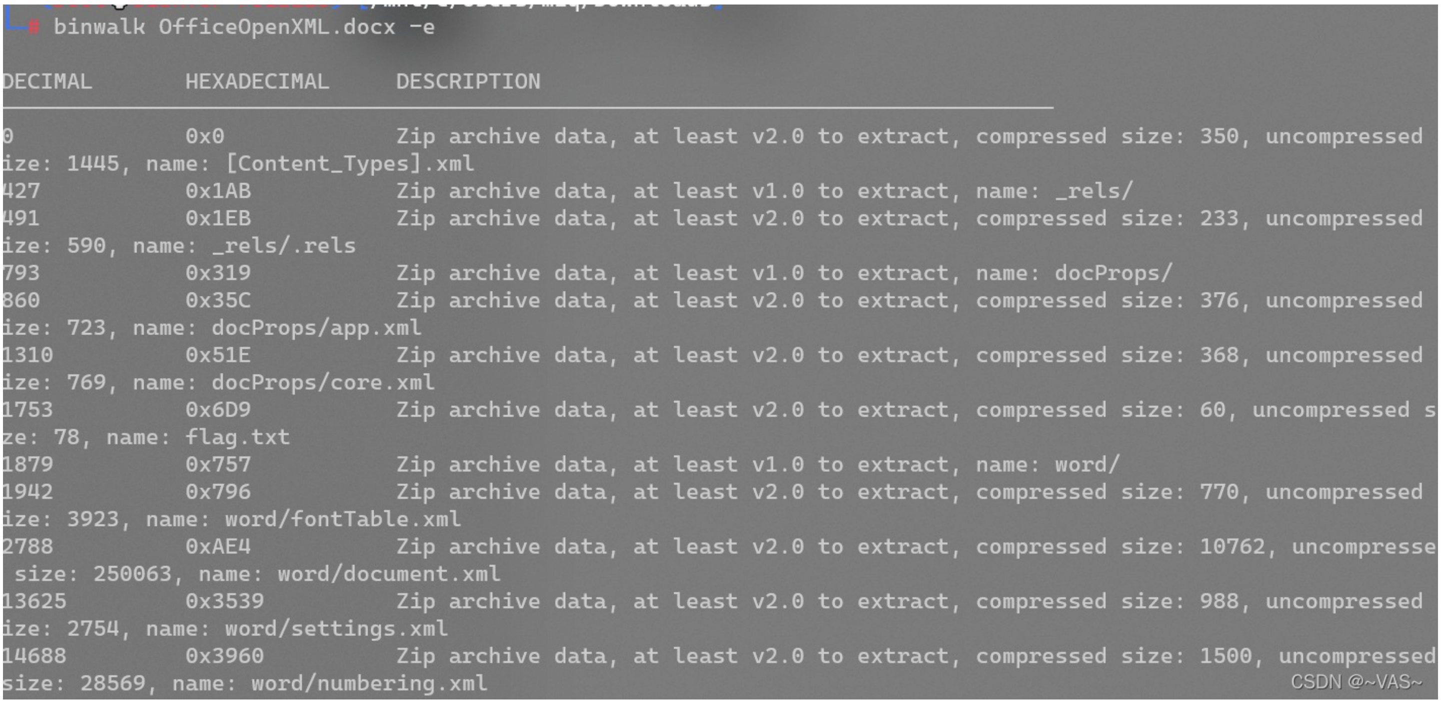
```
@268@>6]U)W)E(&AA<'!Y+"!T:&5N($DG;2!H87!P>2[,
<5VAE;B!Y;W4@;6%K92!S;VUE;VYE(&AA<'!Y+#
G>6]U(&UA:V4@>6]U<G-E;&8@82!L:71T;&4@:&%P<&EE<B!T;V\N
?06YD('1H870@<F5P96%T<R!O=F5R(&%N9"!O=F5R+%
:;6%K:6YG(&$@:&%P<&EN97-S('-P:7)A;"X,
*+2TM+2TM+2TM+0
H5VAA="!D;R!Y;W4@=&AI;FL@86)O=70@=&AE('!L86YE=&%R:75M/U
D5&AA="!B96%U=&EF=6P@='=I;FML:6YG(&]F(&5T97)N:71Y
E=&AA="!W:6QL(&YE=F5R(&9A9&4L(&YO(&UA='1E<B!W:&5N+O
M06QL('1H92!S=&%R<R!I;B!T:&4@<VMY(&%R92!W86ET:6YG(&9O<B!Y;W4N
*+2TM+2TM+2TM+6
<179E<GET:&EN9R!Y;W4@<V%Y(&%N9"!D;RXN+F
<:70@86QL('-P87)K;&5S('-O(&)R:6=H=&QY+G
9270G<R!T;V\@8FQI;F1I;F<@9F]R(&UE+%
=86YD($D@96YD('5P(&-L;W-I;F<@;7D@97EE<RY(
90G5T($D@8V%N)W0@:&5L<"!A<W!I<FEN9^
/=&\@8F4@;&EK92!Y;W4N
*+2TM+2TM+2TM+7
>+RH@2&5R92!I<R!T:&4@96YD(&]F(&UY('!O96TN
B2&%V92!Y;W4@979E<B!F;W5N9"!M>2!&3$%'/R Z*2 J+]
```

exp

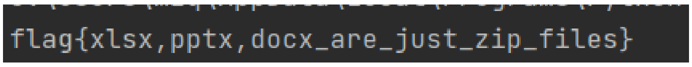
```python
def is_line_contain_flag(line):
    left = line[0] - 32
    return left * 4 % 3


def get_hidden_bits(line):
    left = is_line_contain_flag(line)
    assert (left != 0)
    if left == 1:
        return (line[-3] - 32) & 0b1111
    else:
        return (((line[-2] - 32) & 0b11) << 2) | (((line[-1] - 32) & 0b1100) >> 2)  # 提取隐藏的flag bits


if __name__ == '__main__':
    fin = open("poem.txt", "r")
    fout = open("flag.txt", "w")

    enc_lines = fin.read().splitlines()
    enc_lines = list(
        map(lambda x: bytes(x, encoding='ascii'), enc_lines))
    lines_contain_flag = []
    flag = ''
    for i in enc_lines:
        if is_line_contain_flag(i):
            lines_contain_flag.append(i)
        else:
            continue
    for i in range(len(lines_contain_flag)):
        if i % 2 == 0:  # ????
            flag_chr = (get_hidden_bits(lines_contain_flag[i]) << 4) | (get_hidden_bits(lines_contain_flag[i + 1]))
            flag += chr(flag_chr)
    fout.write(flag)
    fin.close()
    fout.close()
```

Word 文档

binwalk那个doc文件,会出一个flag.txt然后python处理



```
└─# binwalk OfficeOpenXML.docx -e

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compressed size: 350, uncompressed
ize: 1445, name: [Content_Types].xml
427           0x1AB           Zip archive data, at least v1.0 to extract, name: _rels/
491           0x1EB           Zip archive data, at least v2.0 to extract, compressed size: 233, uncompressed
ize: 590, name: _rels/.rels
793           0x319           Zip archive data, at least v1.0 to extract, name: docProps/
860           0x35C           Zip archive data, at least v2.0 to extract, compressed size: 376, uncompressed
ize: 723, name: docProps/app.xml
1310          0x51E           Zip archive data, at least v2.0 to extract, compressed size: 368, uncompressed
ize: 769, name: docProps/core.xml
1753          0x6D9           Zip archive data, at least v2.0 to extract, compressed size: 60, uncompressed s
ze: 78, name: flag.txt
1879          0x757           Zip archive data, at least v1.0 to extract, name: word/
1942          0x796           Zip archive data, at least v2.0 to extract, compressed size: 770, uncompressed
ize: 3923, name: word/fontTable.xml
2788          0xAE4           Zip archive data, at least v2.0 to extract, compressed size: 10762, uncompresse
 size: 250063, name: word/document.xml
13625         0x3539          Zip archive data, at least v2.0 to extract, compressed size: 988, uncompressed
ize: 2754, name: word/settings.xml
14688         0x3960          Zip archive data, at least v2.0 to extract, compressed size: 1500, uncompressed
size: 28569, name: word/numbering.xml
```



```
flag{xlsx,pptx,docx_are_just_zip_files}
```

三教奇妙夜

```
import cv2
from matplotlib import pyplot as plt

file = cv2.VideoCapture("output.mp4")
ret, preframe = file.read()

while True:
    ret, frame = file.read()
    if ret == 0:
        break
    diff = cv2.absdiff(preframe, frame).sum()
    if diff > 10000:
        print("diff: {}".format(diff))
        plt.imshow(frame)
        plt.show()
        preframe = frame
```