

DDCTF 2018线上赛writeup

转载

weixin_34077371 于 2018-04-24 14:35:00 发布 99 收藏

文章标签：人工智能 python json

原文地址：<http://www.cnblogs.com/semithigure/p/8930257.html>

版权

第一题：



d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9b2b2e1e2b9b9b7b4e1b4b7

解题思路：

首先尝试各种解密，无果。

开始研究，分析字符串，首先看看有哪些字符，然后准备分析频率

```

import string

cipertext =
"d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fb9b2b2e1e2b9b9b7b4e1b4b7e3e4b3b2b2e3e6b
4b3e2b5b0b6b1b0e6e1e5e1b5fd"
for i in string.lowercase:
    if i in cipertext:
        print i
for i in string.digits:
    if i in cipertext: print i

```

打印出来发现字符串由a-f 0-9组成，瞬间想到16进制格式，开始向16进制进攻

len()查看到字符长度为134，16进制一般是两个字符组成一个字节，所以两个两个拆解分开试试

```

cipertext =
"d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fb9b2b2e1e2b9b9b7b4e1b4b7e3e4b3b2b2e3e6b
4b3e2b5b0b6b1b0e6e1e5e1b5fd"
i = 0
plaintext = ""
while i < 133:
    plaintext += str(int(cipertext[i:i + 2], 16)) + " "
    i += 2
print plaintext

```

得到212 232 225 244 160 247 225 243 160 230 225 243 244 161 160 212 232 229 160 230 236 225 231 160
233 243 186 160 196 196 195 212 198 251 185 178 178 225 226 185 185 183 180 225 180 183 227 228 179
178 178 227 230 180 179 226 181 176 182 177 176 230 225 229 225 181 253

明显看到超出常规ascii范围，考虑尝试凯撒解密，由于ascii有128字符，尝试先减去128

```

cipertext =
"d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fb9b2b2e1e2b9b9b7b4e1b4b7e3e4b3b2b2e3e6b
4b3e2b5b0b6b1b0e6e1e5e1b5fd"
i = 0
plaintext = ""
while i < 133:
    asciinum = int(cipertext[i:i + 2], 16)-128
    plaintext += chr(asciinum)
    i += 2
print plaintext

```

得到flag

第二题：

提交答案

已解决291

X

第四扩展FS

100

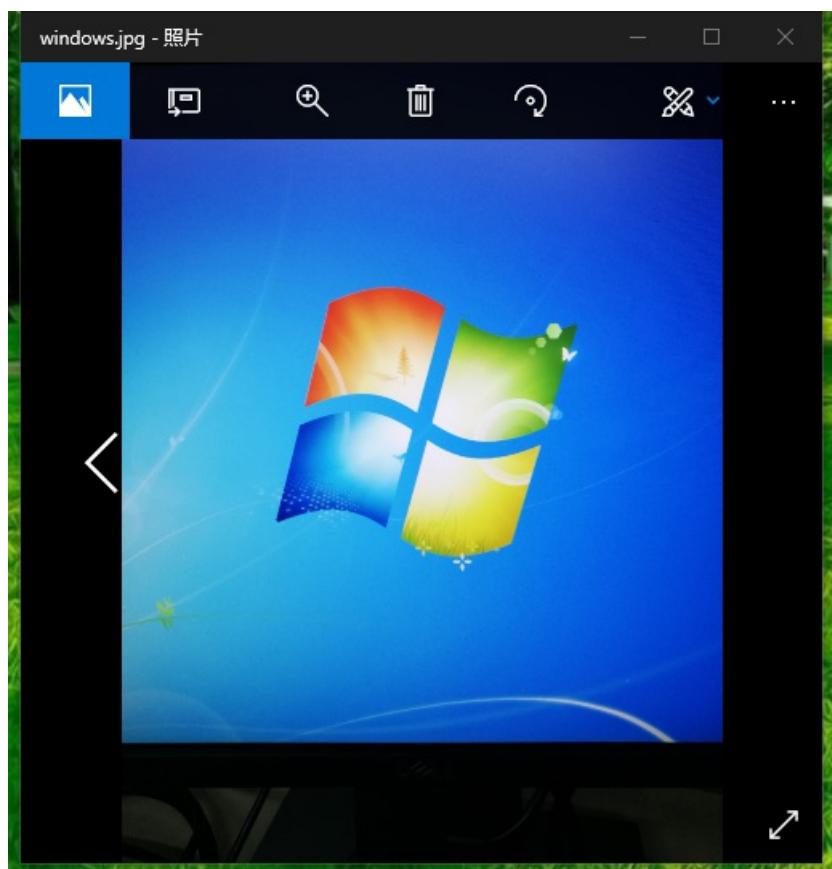
D公司正在调查一起内部数据泄露事件，锁定嫌疑人小明，取证人员从小明手机中获取了一张图片引起了怀疑。这是一道送分题，提示已经在题目里，日常违规审计中频次有时候非常重要。

 windows.jpg

Flag

提交答案

打开下载的windows.jpg



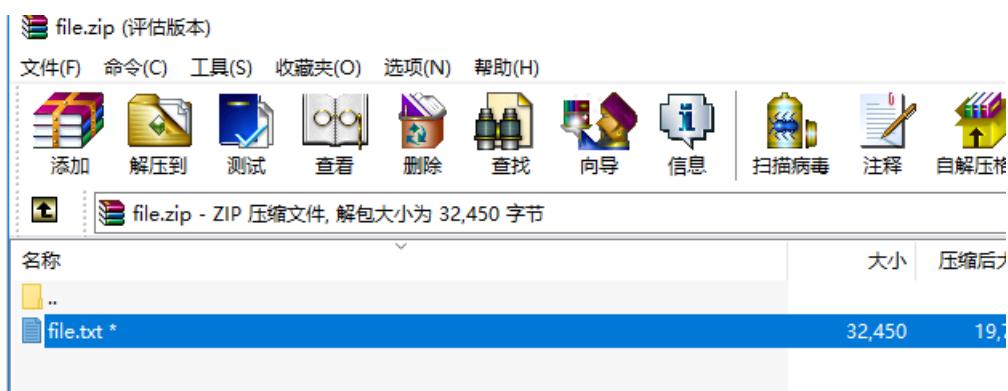
非常普通的一张图，看到图片，第一联想到隐写术，自行用binwalk看看图片有没有藏东西

```
Windows PowerShell
PS C:\Python27\Scripts>
PS C:\Python27\Scripts>
PS C:\Python27\Scripts>
PS C:\Python27\Scripts> python .\binwalk C:\Users\fuzhi\Desktop\windows.jpg

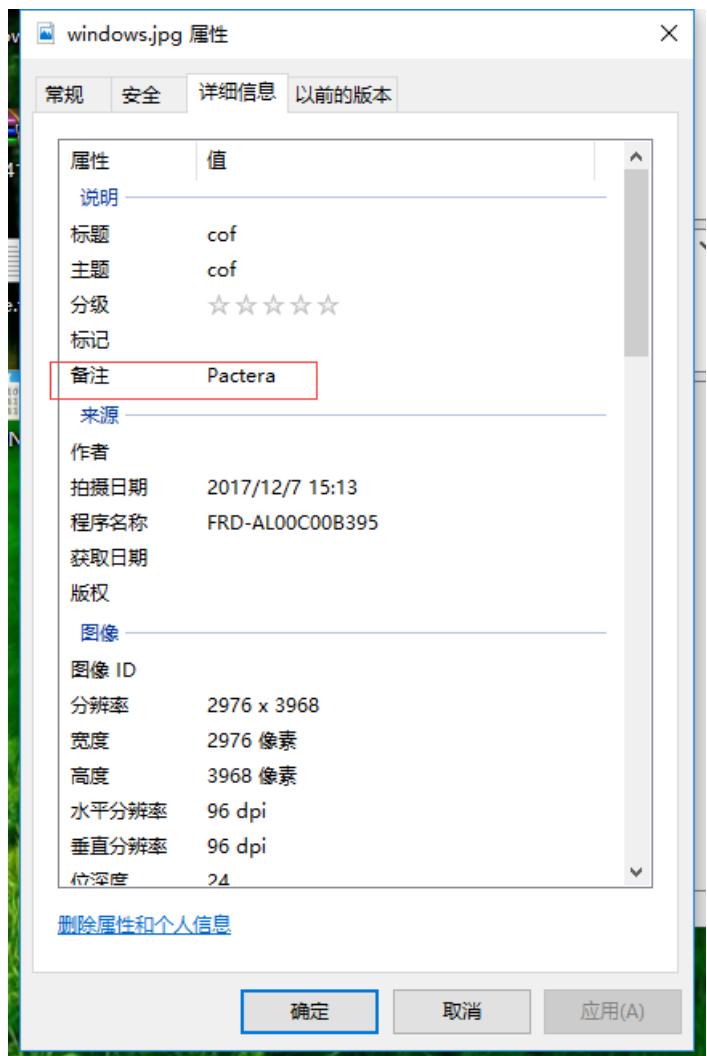
DECIMAL      HEXADECIMAL      DESCRIPTION
---          ---          ---
0            0x0          JPEG image data, JFIF standard 1.01
30           0x1E         TIFF image data, big-endian, offset of first image directory: 8
5037          0x13AD        LZMA compressed data, properties: 0x64, dictionary size: 0 bytes, uncompressed size: 16777
216 bytes
2825118      0x2B1B9E      Linux EXT filesystem, rev 2.0, ext4 filesystem data, UUID=57f8f4bc-abf4-0000-675f-946fc0f9
c0f9
2872753      0x2BD5B1      LZMA compressed data, properties: 0x5A, dictionary size: 0 bytes, uncompressed size: 17179
8757376 bytes
7236510      0x6E6B9E      Zip archive data, encrypted at least v2.0 to extract, compressed size: 19743, uncompressed
size: 32450, name: file.txt
7256345      0x6EB919      End of Zip archive, footer length: 22

PS C:\Python27\Scripts>
```

一大堆东西，尝试提取分离，得到一个zip，打开一看需要密码



这里可以用暴力破解，不过花费时间较长，期间可以做一些其他事情，比如去看看图片的详细信息(可能藏东西)



在备注找到Pactera，尝试作为密码输入，解压成功Orz.....

打开file.txt

file.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```

|0hZ4huHCjujii1 {stzenozuFDTatOsj1HzeuDzs4T4tDaz1CiOT4oa0j11juuijeHfCah} jeDH4e {DD+D1DCDaTCThzH} FsFntn4a1
|{aCu1DtDjHD1DDCCbCiejsDHi144zThjzF4h} 1FhehtTiFse44HTC0z {Tn0t{u} enh} jsnet} DDTduuzziaHH0s{jC4D1D41e0H} 1j0aH{eD
|{CeICn1DuatDjotDFhzFTdFhehtTiFse44HTC0z} Tn0t{u} enh} jsnet} DDTduuzziaHH0s{jC4D1D41e0H} 1j0aH{eD
|{DzDDtneOCDnCeDCzhejDf4s} {tnDCHt1DCTD1CDCFTD} a4hFijjuadTf0t4TujhTiC1} sD0bPFtCjCDDoi1 ouhHTau1D{jC} OF1HFD1DunDFnD
|{OH0hCDls14HToFnTl1elzoTThmfa} tto} Djt{C} hCet} ihanul0DjeF0juWFszDhnM {HhhhF} {senDn} {DCDs4FheFHdzHTHta1TetTCeboDeCCee4H4neTn1tCn
|{sFzCss} {tals4eltuaih4DHoTuoDjenhC} FaF {DoetaihiTH} D {DiDCi} HuTju DeCsTDe {t} DHDDDSuhsuPoisHeDjeHjtDip} 1jCh{DF} {0{aDn
|{1sCFDejFDC1lujozTTeCiaeHajiZlusc} {zjc{DmDiss} {z1ChRut40zTuiDe4t0aTjCd1ThC4teDztu{jDHC} j} {THhH} {CCuFsnDeH4zDje}
|{110HCFCsDFFri{jC1D1H4HCDOobhhHF} dHFD} {nDnlb1t} {zjTDCeCnHAd} {te0Ti} {TodifDfihDaa4DastTiiOhofnsD4zTDD} jhe0oitTtumHiTilFioj
|{14itFzaDDchoi0hHie} {o{14D4T0Co4juouWHSwulCioo} {F0nCjiuDuTuel1TDFCCin1tDuh04htD} {T1hj0tt1HaOn}
|{s}j0TD{jnsitCD1thD0eiszonbTsnjeuDchDtsuei4nD} {u{4nTdjDCsC} t} {s}HeaeeDdeCfiC{T} {i4DdTTHHDH}
|{0oi1jhzbzT10PhiaDh4jF1t1haFzN0sts1Hht0niDFCT0uW4jTfZHDuFuT1leDe} {enFFejj0CjC14tH} {tuHteam} n1Cs4tj1TH {jHuhdHD0hC0FnDs} je
|{e}fCtHuEn10eekhuT4C4eeoo4DCTe4CtTuThb {1{h}Ce1Fuh1jTlt} {F4D} {FD0} {u0ozCuTC1s4hnDssnFzTTD{jzTC} jhCdtlat {zD10Cooohbj4{De1} Hj}
|{F1tzt0DFD1aDzutsDjCnH4t0440nln{0s1Hh} Os} {HDC40414thjsh} {o{lo}} z0i1CsfzuHc1Cj0ohDht1 {atDaa} {1D} {t}HDCO} aDD {CtFC4z} {Du0DiaeToH} an0 {T4nuFz
|{DHhHotTTD} {t}FctDHCtC0jz1hF0oT {1ta00jt} {{edi} {i{4{s}t} e} {z}zDn} {zlotDCeun4Dzoh} {i{j}Hfez1jTzndi1tFf1j} {TaeFh1} {oFhuTOHCd1D0CeCoc11
|{CCjjuuDFC00u1T1s1djuua1jH0Te} {1e} {ToiuuunoCjutC} {u{j}halCsz4ujFduo0404iul} znT {4DT} {et} {s}jDnDj1tTzzjsC1D0 {u}D0T0oosD
|{D}jDzns0CzZCFFeCasTenId0e4DFD0s0us} DeneazT {F1F4tnz4Cstj10DH1} {DFjHeTa0zi0tzFFCDuFo40zr0Ch200tDdtCassFCeCDFCTCjnDn0
|{e}HOzijD0aTs1tTzhlaFDDCDFjejDHTHniaeTFDF4CzHoD {ejaC} {Due} {thD0HanisahsD0TFH} utsCCD0TeT41 {DT} {DneDje1DdUTDTHone
|{Ch1DtniCs0iD0FzsTcnt0j0TDFCT} {j}HoF4tCT {HaF} {az} jHsFsehHis} {Tt} {t}Hah4D {z0e4nDw0} {Dde} uD0Dii0uWTh0D {s}4t4a4szjDz1FDHtnen} hHsei} osu1C
|{hheFOODD} {t}huCh1l0uDaDh04jDj1hDuh1Dul {Ct4} {t}zstD {i{1h} usuusz4azOF} sTT0ehDz40uFdhfPCHD {Dl} zhseCjD0TjDz1HjhnsT1aTleHsFjC01FnD} DDHauo0h
|{CFHhDDDjnf0eDfTj0aTizjH0oe0uHieOnf {zD} {DC} {1j,j} zuiulnsLh1T4DDDjzHfHsuhuauTDDHtTnunt {i}CeuOchniDTo} {C}heeoHTlionh14HohjDD1 {D}lieult4F
|{uiDtnHTD} {1{f}hDHCt1ToDouujzTszuTanuttjDcdj} {0} {4T11} C0zj0DhDits1C} Tztsjia4TzHoeDDe4e1jTzDhhejCH {ae} TdA {u}dsiDF4Dz1ehjs} 4DFTfCtC4s0
|{juTfhuuui} {D}hiosh {C}lanid4s4t0H0h{j}j0} {HCFi1Cjte} {l}s1s1FzT1aD4hTf1jHsHe44jC1jDohe4ztoTuHj4CzofcoThe4DDzuz} {D}tmhzD1FT44FDt {Dn} {hsCzahTdsi}
|{010H} {1}bneDCCnsf1l0s1Fc0j} {s}zCzuo} {Tnaoj} {C}hihFoe} {h1} {D}j{j}Dc4a0} hioznoD0t} CTTazszuTsz1uho} 4cTTjez {nt} o} arThIn} soej0TaT1jith}
|{jjoODDsFhuOHO0} {zD} {t}lssOnFFt0DZeettj1iT} {Ta} {j}DeFiT4aCj40} H11ia4DHiuh4uoTiD4DsD1 {Htd} {s}0iaeTj {j}DCOTtosozit
|{Hf4eHF0D01Cf0s1tACD0z1FzjDjFt0} zBeTheHTFThFdTs {1{e}idt4FzTfutD04a4} {j}ahjDtsiD0tHTT1ssFDFFODes {i}H4} l1t0Cu4a}
|{1tsjeCHTD1Hj10zn} {u} {e} {H}Hj1s} {t}HFCuCD {u} {O}FOj {zD} {D}Ch} {i} {H}D {D}HCCenDneH04DDTdiH1C010hC0CzjH1z1Df1ea0sCh {C}te} C1HsFuFaijFa10zCdo
|{T}f4D1z4s14HeCheTHe} {s}iztC {u} {j} {j} {s}osDDz2i1a {u} {z}t0hs {j}DHCa0nC {o}i1fua4C} {D}sd {H}bhFuT01Ddahno40nDf4e0a} 4ThtDj0hAeC1HFtTte
|{T}b1ODDDhazFzheFhsT {C}tznzshFnsFF {H}jui0 {e}hCt1lis1j1Ciasannmuo{j}CzD {e}HnTzFtDhszjTu} {O}ieHb0 {i}D4sFjuHzh {e}TuiiaTuiFauT} tDQFiddF0D4t
|{D}1s1jDhDhazj0eC {j}njjnTFCueabFnDjDFH1alD1D {e}rjDsFstH {H} {D}sdDf1Dztdzoua} {D}CHTf0u {ee}04CshCjTz0C1u0D{j}Cte {o}FzFz} ta4jsu
|{unjhshsCuChtDuC04Htj4TiltsajCFC {i}CtDTh2o1FtDzta {e}b} Ch4z0zlndHs1ODDjFutjFfu00O} OOH} C {u}zDunCDD1hba {s}THsTDFj1s1tD4CahsTn {0}j4
|{es}TaDeeCaCHC1TfjDanuCTD {h}onaijn} {j}hjTisoTH} {} {he}0jDDFCoAci} suTDDjjsFe {1C} {e}DCjF4a0 {t}el {ee} {FFCuHCaeiheina0hu}j0TDD {1u04} {es}
|{D}j141hHDfHse1eToaDfjHjTb4e0D {Cu}BuCuT} sHHC {t} {e}D} {0nsDuzTts1HfaHizzhoiDHzoHtDHD1DCH4s1aD{s}etThjFD0o} sFabsjFDj0atjjo {F}o {j}njHtz1
|{FDFFChDes} {T}oF1D1DC} {C}44z1DDDDojaauas {z}tj1 {n}T11et} {C}1j1FhtH4jDj0sCh {H}eF} DF {s}dU10h1TTCjDfsj0sDeDj1tFuD1s} HciTiDtiD4FFChjA
|{h1nautC} {h}ciC {C}1C6CtEfh10DztsDD {C}114CsetTosDh101hmDuoCoDuTi} {t}tchDh1jnt4D1s1T0s{j}CtaunoHzz {niu}TTthj
|{D}ts40iDjse0j1H1DcDtzjDjDOCtuozzHtFCDDF1D1t0Ds1 {s}t {D}h {a}fzFFD {D}CoCDDjT4ao} {ne}CfidoHsHOT} 4FToFzj1h0T {000Dn1} T4eo00D0 {t} {T}dai
|{h}j1TidTea4FnsD} nTui1T1H1o} {C}z4tFuehse4HusFfjeCho4Hul {s}tnzDe1a4s2Cut4hjsColtzDcsDn1 {D}Cs
|{m}0o0DhTCTe4C0DC{j}deTHD0HtF1FnheoTadT4i {t} {T} {D}hTfHfjuW1ut} {e}FDjedWdUe0TsF {T}jz} {C}isj0H01H1HfC1eTDFono} {z} {4CFa} {D}
|{ss}DD1h1uBsDfHdeauajhs0T4CDon {t} {e}D} {s}Tfje1s1DeatHeDd0t {j}0Fbj1TdbHo {ut}zHd1eejjuT {HHfajFtenajufDhsnCeemin} {i}De1s1
|{ue}0u1jje1jCtDjDFH1D0o0ztu {H}fsD1 {o}tiaD} {uo}CeaZsj} {O}caui {C}FuC {0s}4naFCzhaeCOF} {n} {z}DFCn1 {e}1tt4j1j1HTCTDcue} {n} C4 {e}t40tizheud0t1DDnFhz
|{1}CdfoujC {ah}COPFeth} FDCCzTHD} HdcE {z}HH} eeCDDei {H}stT1TTzesmuFzDeHeszzCujtsTTDFt1o1TCDCD1D1Ct {To}CaoC} Hsdaaue1D {e}4eDu {1j}Do
|{e}4DaDeeaTj1HzzuHhze} 4h {o}HDhmeHD1CTTCD4thF} {D}4} ja0ia1jTuu0iTOHn4 {s}d1aCj1D1j1hDj1F1FFu0DD11 {e}D1FzauHe4s1i0F4enDtD4 {0}oeF0atD1D
|{ia}f1nHnFfetj1o2DnsjD04DFiHeDsTf} jiFeia1hs1aHFzFDDj1D0oNtDhszC0D04Hc04Cj1i1Teazj {i} {i}C4h4aTTTjtuaszDhas {D}nei {n}H{j}ii
|{es}DaeC1hahF} {a}Toi} aDs4C1h1 {11} DfHh {C}ielh {FD} {n}tsH} {eh} {z}4teCfA {s}1j1Th4s {z}dej14a{j}DtC00eCDoChaleHCDnjsstDfisFhTodin0FjDDe4t {D}HtH
|{t}0n01suhhazF} TntCCsan04uTCD1FC {u}C {s}C1CTu} FCoj {ju}Hai1tztzFjesanhTsDotjzFLITFDT1iH4o1C1f4e{i} TDDFa0jex {0u}nD1Tds1D0} zj1tFzHoDTC
|{FT4tA} HjiojD4} } untH1tDzjDuaFnCjTnj4zjejejuH4u14Dtn1te0 {C}hoTfC0t au1tC} ujDOD1o4CoHnDaoDs0HzuH4Tj1telnF1Tj} CDCjeeiFz0tD} jT4
|{D}11Hj1Hc {C}hfdTtus0 {el}hTt1aejCCO} seDHe1aChCnSco1} o}I1s {t} {C}DNTD0 {HTt10D1eo1s1jFtCDFT1t0i4TeHdZ1lhbtde1a} {o}Ts0TjjaAtDz0DHD
|{F}iFHO0TDCjnit14e1b4aTbDThnD4zazTFTai0aH1} DhH0 {h}j1 {j}si11eeheahzeDe {D}tijCFhHD0D0ei {e}DncojeDFzz4DTh4j1C {C}CHt ueeDDDu4jD4zCiuaj1
|{hzssu1} {t}1i14 {C}tiu4 {a}1 {l}1 {j}Chza {u} {o}DC4TFF0eSu} {ju}T {u}tsz {D}fE} {sn}j0sCD {u}D1TnHt} {s}0DCsnHjuT {1}D0s11 {T}sDan4D01mutDaTh4s
|{h}j1s1jDtsDtuDThD04TDes {T}nHF0C40FhsFzeiDHH0D {4}z {4}z {s}zjucouFl1luFjha14F1sDz} {t}00FTOC} {s}0 {DD1DeHzCf1uDeheaiTseD} {Fs}DeeCenD {t}
|{i}FDD {C}o{j}Dd1 {4}zienBnDDDDH {D}Hh0 {sz}4chTOD} {o}4DeoDzTjHjse1jC0 {C}d{j}TD1FbSsejDjs {itt}D44D {t}iui0habHzuD} {D}j1u1ljoC1f1s44} {D}s {T}H
|{1}TnD4ohD4hDjciuhtFta1TjCt4t0eH4z4DCCzF4juH0j1OD {T}D{j}De1ztnH4CtaF {D}n1sn4} {e} {a}1CHsn1TfEfa00eaDja {F}DTH4DFTeDDD} t} hnje
|{1}shTCi1usuoza0hTu4j0o0sFzDeuF1FzidDjess {s}iozofTuDjoeDohtz4uF4 {T}uC10szsT {C}fnsjDhHFjasaDj1oDje1D4TCDH0n1tsCh
|{FCs}D1OHuhejeDTjDnTsF14a0nC {is}Dse14T400TTj} {i}1auj1Dj {D}eDzTc4z10ln} {ou}D4i} {o}4CaFh4eHDsee {t}1 {j}0Dd1t4JFj01CseunF4uCj {az}Thz
|{ihu}FaDDzHuH4i0iCC1LHTdanjzn1lujnDj} a1ieDH1D {o}0oh1oChTH1ezai1z04etCj} {t}ta1} {o}F} 4hzuFizTDCj {e}F0 {DT}44CtCCFCDaCj1tCDH01Ht {Founaj
|{inh}0jDteFDjuuFtsF0iZh1stDF00ta {Hu}Dinun {D}Doco0ta0HDD} {H}hue0ujFDFhCDDj} {h}chsDh1a0Te4u1jeaCD4TuFFoiFlujoCFSjFni} {D}jTCF4 {ts}T

```

进行一帧栅栏，编码等解密，无果。

破了很久，最后回头看题目看了的提示

D公司正在调查一起内部数据泄露事件，锁定嫌疑人小明，取证人员从小明手机中获取了一张图片引起了怀疑。这是一道送分题，提示已经在题目里，日常违规审计中频次有时候非常重要。

尝试分析字符频率，然后进行排序

```
file1 = open("file.txt", "r")
cipertext = file1.read()
dic1 = {}
for i in cipertext:
    if i not in dic1:
        dic1[i] = 1
    else: dic1[i] += 1
z = zip(dic1.values(), dic1.keys())
plaintext = "" for i in sorted(z):
    plaintext = i[1] + plaintext
print plaintext
```

直接得到flag

第三题：

提交答案 已解决136 ×

流量分析

200

提示一：若感觉在中间某个容易出错的步骤，若有需要检验是否正确时，可以比较MD5: 90c490781f9c320cd1ba671fcb112d1c
提示二：注意补齐私钥格式
-----BEGIN RSA PRIVATE KEY-----
XXXXXXXX
-----END RSA PRIVATE KEY-----

1cMYNO2T...

Flag 提交答案

本题差评！！！干扰项太多，不亲切，这里不再赘述踩的坑

下载得到附件，是一个pcap文件，果断丢入wireshark，把解析的到的文件一次导出

只有IMF的能导出，分析查看，大概在一个文件比较大的地方，看到

--B_3598538194_700708737
Content-type: image/png; name="image001.png";
x-mac-creator="4F50494D";
x-mac-type="504E4766"
Content-ID: <image001.png@01D38B05.8079CE40>
Content-disposition: inline;
filename="image001.png"
Content-transfer-encoding: base64

iVBORw0KGgoAAAANSUhEUgAABIwAAAISCAYAAACu61SmAAAMKG1DQ1BJQ0MgUHJvZmlsZQAA
SiMvVwdYU8kWnluSkJDQAhGQEoTpVepoUUQkCrYCEkgocSYEETsyKKCa0HFghVZFVFWLYAs
NixYWBTs9WFBRVKXCzZU3iQbDfV7733vfN+5979nzp5z7kz880AoB7NEYuzUA0AskU5kpjQ
QObEpGQm6SFAArIwBM4crhScUB0dASAMvT+p7y7Dr2hXLGXx/q5/b+kJo8v5QKAREOcypNy
syE+BDuxhVLcgAg9EC72cwcMcREyBjOsSyBBiM3l0F2JPeQ4VYkjFD5xMSyIuWBgQxi4knQA
10S8mLnccBhHbRnEdiKeUARxE8S+XAGHB/FniEd1Z0+HWN0aYuvU7+Kk/yNm6nBMD1d9GCtz
UYHkKFaqzuLM+j/L8b8100s2NIY2VKpAEBhYz1let8z4p4XJMhfcKDUyCmItiK8KeQp/OX4i
kIXFD/p/4EpZsGaAAQbz5XGChE2gNhU1BUZMWj3TRGosCGGtUfjhDnsOGVf1CeZhjMYH83j
S4NjhzBHoohl71My4wPGIy5Rcb8DVsztFeJSp5opzdQhZmREKtBfFeeGrs+6PM8X8CKHPKR
yGLknOE/x0CaJCRG6YOZ20uHs8kBEJ25CCOyBHehSn7Y105HAU3XYgz+NKJEUM8efygYGVe
WAFFD/IHysV5wIGDPpXir0IB/2xJn5WqNxuCnGbNDd2qg9vDpxsynxxIM6Jj1NywUzoooo
1RxwWxABWCAlMIEmaiqYDjKasK2nvgd+KvtCAAIdQDrgA/tBy1CPREWLCd5jQT74CyI+kA73
C1S08kEuTh8Ztjqf9iBN0Zqr6JEJnkCcDcJBfvyWKXqJhkdlAI+hRfjT6FzINQuqv0OnG1N9
yEYMJgYRw4ghRBtcH/fFvfEI+PSH6oR74J5DvL75E54Q2gkPCdcInYRb04QFkh+YM8F40Ak5
hgxml/p9drqlj0qKB+i+MD6MjTNwfWCpu8CRAnA/OLYrtH7PVtac8bdadSYi05BR8giyP9n6
RwZqtmcuw1Hk1fq+FkpeqcPVyG23/JghH67v68eA7/EdpBa12EvgtBtmLnsSasHjCx4lgDlood
1ePhufYMTeGRotR8MmEcYQ/jccZHFNeNa1DtUo3w+fBnpDDz8uRLxbWdPesiBdKmmGls1
n0kWcUePYjo50MJdVL73K7eWnWzFno4wLnyzFewAwMd3Y1Gcg62stvAuAg70AU059slnDajw
AcC5NVyZJFdPw+UPAqAAdbhS9IAR3LusYUZOwA14A38QDmaBKBAHksBuWGcBnKsSMBPMAQtB
Fg=PK8F=eBFERTw+bv+RPHAD1AmcR/GFBPvA2VAN34F7z+q1Q13cH4NFFTSF0hT7oTceTBWkH

复制出来，看到是base64，解码，得到一串字符

Base64编码或解码结果：

这个看到应该是一张图片，所以这里保存成图片看一看。。

```
#!/usr/bin/env python
# coding=utf-8
import base64

file1 = open("test", "r")
cipertext = file1.read()
file1.close()

plaintext = base64.b64decode(cipertext) # 读取file1进行base64解码
file2 = open("testpng.png", "wb") # 以.png写入保存
file2.write(plaintext)
file2.close()
```

打开我们刚才保存的图片

MIICXAIBAAKBgQDCm6vZmc1JrVH1AAyGuCuSSZ80+mIQiOUQCvN0HYbj8153JfsQ
LsJIhbRYS7+zZ1oXvPemWQDv/u/tzegt58q4ciNmciVnq1uKiygc6Q0tvT7oiSTyO
vMX/q5iE2iC1YUIHZEKX3BjjNDxrYvLQzPyGD1EY2DZI06T45FNKYC2VDwIDAQAB
AoGAbtWUKUkx371LfRq7B5sqjZVKdpBZe4tL0jg6cX5Djd3Uhk1inR9UXVNw4/y4
QGfzYqOn8+Cq7QSoBysHOeXSipztW2cL09ktPgSlfTQyN6ELNGuiUOYnaTWYZpp/
QbRcZ/eHBulVQL1k5M6RVs9BLI9X08RA17EcwumiRfWas6kCQQDvqC0dx12wIjwN
czILcoWLig2c2u71Nev9DrWjWHU8eHDuzCJWvOUAHIrkexddWEK2VHd+F13GBCOQ
ZCM4prBjAkEAz+ENahsEjBE4+7H1HdIaw0+goe/45d6A2ew0/1YH6dDZTAzTW9z9
kzV8uz+Mmo5163/JtvwYQcKF39DJGGtqZQJBAKa18XR16fQ9TFL64EQwTQ+tYBzN
+04eTWQCmH3haeQ/0Cd9XyHBUveJ42Be8/jeDcIx7dGLxZKajHbEAfBFnAsCQGq1
AnbJ4Z6opJCGu+UP2c8SC8m0bhZJDe1PRC8IKE28eB6SotgP61ZqaVmQ+HLJ1/wH
/5pfc3AmEyRdfyx6zwUCQCAH4SLJv/kprRz1a1gx8FR5tj4NeHEFFNEgq1gmiwmH
2STT5qZWzQFz8NRe+/otNOHBR2Xk4e8IS+ehIJ3TvyE=

这里就要使用图像识别了，ocr了解一下

```
#!/usr/bin/env python
# coding=utf-8

import base64
import pytesseract
from PIL import Image

def getcode(imgurl):
    """识别图片"""
    image = Image.open(imgurl) vcode = pytesseract.image_to_string(image) code =
base64.b64decode(vcode.encode("utf-8")) return code temp_imgurl = 'testpng.png' code = getcode(temp_imgurl)
print code
```

这里无敌坑，ocr识别准确率不是很高，code得到后还要和图片进行比对检查。

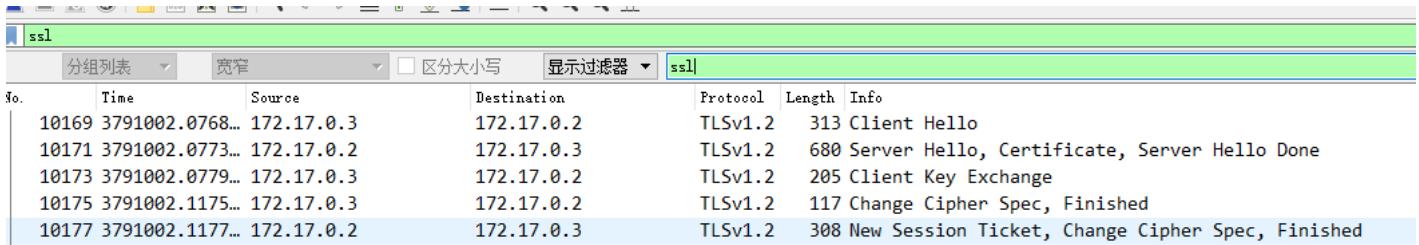
根据提示，补全RSA

-----BEGIN RSA PRIVATE KEY-----

MIICXAIBAAKBgQDCm6vZmc1JrVH1AAyGuCuSSZ80+mIQiOUQCvN0HYbj8153JfsQ
LsJIhbRYS7+zZ1oXvPemWQDv/u/tzegt58q4ciNmciVnq1uKiygc6Q0tvT7oiSTyO
vMX/q5iE2iC1YUIHZEKX3BjjNDxrYvLQzPyGD1EY2DZI06T45FNKYC2VDwIDAQAB
AoGAbtWUKUkx371LfRq7B5sqjZVKdpBZe4tL0jg6cX5Djd3Uhk1inR9UXVNw4/y4
QGfzYqOn8+Cq7QSoBysHOeXSipztW2cL09ktPgSlfTQyN6ELNGuiUOYnaTWYZpp/
QbRcZ/eHBulVQL1k5M6RVs9BLI9X08RA17EcwumiRfWas6kCQQDvqC0dx12wIjwN
czILcoWLig2c2u71Nev9DrWjWHU8eHDuzCJWvOUAHIrkexddWEK2VHd+F13GBCOQ
ZCM4prBjAkEAz+ENahsEjBE4+7H1HdIaw0+goe/45d6A2ew0/1YH6dDZTAzTW9z9
kzV8uz+Mmo5163/JtvwYQcKF39DJGGtqZQJBAKa18XR16fQ9TFL64EQwTQ+tYBzN
+04eTWQCmH3haeQ/0Cd9XyHBUveJ42Be8/jeDcIx7dGLxZKajHbEAfBFnAsCQGq1
AnbJ4Z6opJCGu+UP2c8SC8m0bhZJDe1PRC8IKE28eB6SotgP61ZqaVmQ+HLJ1/wH
/5pfc3AmEyRdfyx6zwUCQCAH4SLJv/kprRz1a1gx8FR5tj4NeHEFFNEgq1gmiwmH
2STT5qZWzQFz8NRe+/otNOHBR2Xk4e8IS+ehIJ3TvyE=

-----END RSA PRIVATE KEY-----

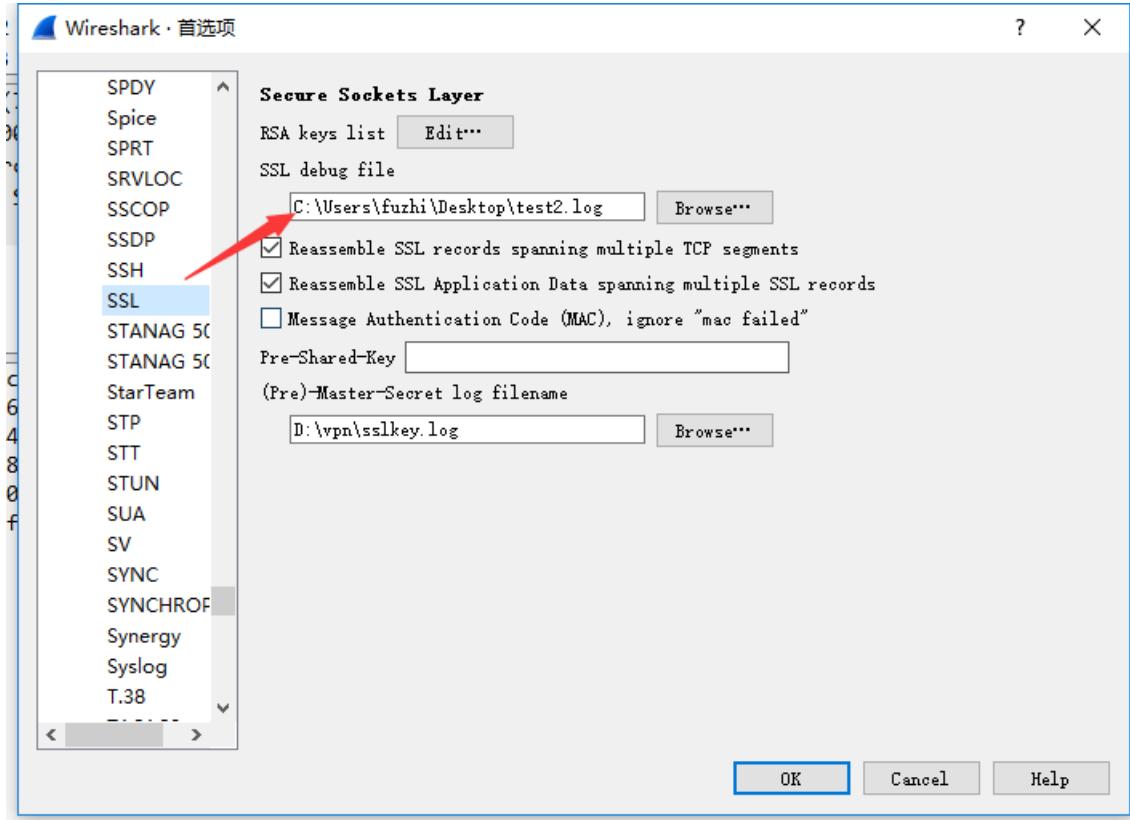
看到这个rsa秘钥，自然联想到ssl，搜索一下（ssl了解一下）



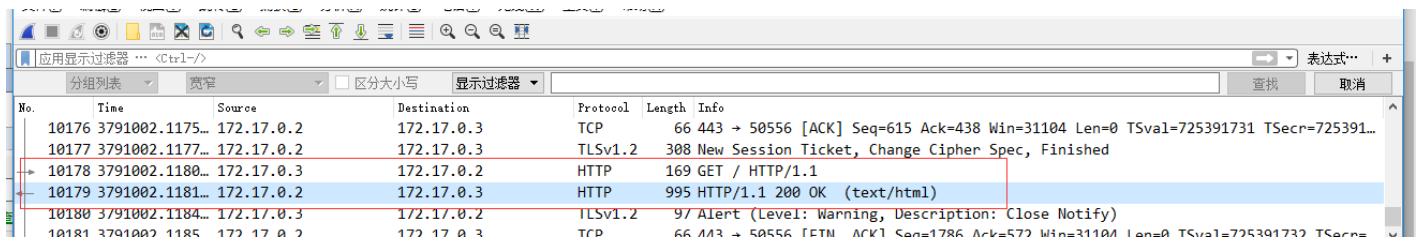
No.	Time	Source	Destination	Protocol	Length	Info
10169	3791002.0768...	172.17.0.3	172.17.0.2	TLSv1.2	313	Client Hello
10171	3791002.0773...	172.17.0.2	172.17.0.3	TLSv1.2	680	Server Hello, Certificate, Server Hello Done
10173	3791002.0779...	172.17.0.3	172.17.0.2	TLSv1.2	205	Client Key Exchange
10175	3791002.1175...	172.17.0.3	172.17.0.2	TLSv1.2	117	Change Cipher Spec, Finished
10177	3791002.1177...	172.17.0.2	172.17.0.3	TLSv1.2	308	New Session Ticket, Change Cipher Spec, Finished

看到了ssl的通信

把秘钥导入



重启wireshark，惊奇地发现在最后多了http



No.	Time	Source	Destination	Protocol	Length	Info
10176	3791002.1175...	172.17.0.2	172.17.0.3	TCP	66	443 → 50556 [ACK] Seq=615 Ack=438 Win=31104 Len=0 TSval=725391731 TSecr=725391...
10177	3791002.1177...	172.17.0.2	172.17.0.3	TLSv1.2	308	New Session Ticket, Change Cipher Spec, Finished
10178	3791002.1180...	172.17.0.3	172.17.0.2	HTTP	169	GET / HTTP/1.1
10179	3791002.1181...	172.17.0.2	172.17.0.3	HTTP	995	HTTP/1.1 200 OK (text/html)
10180	3791002.1184...	172.17.0.3	172.17.0.2	TLSv1.2	9	Alert (Level: Warning, Description: Close Notify)
10181	3791002.1185...	172.17.0.2	172.17.0.3	TCP	66	AAR → 50556 [FIN ACK] Seq=1786 Ack=570 Win=31104 Len=0 TSval=725391732 TSecr=

追踪http流

Wireshark · 追踪 HTTP 流 (tcp.stream eq 2019) · 1cMYNO2TCQHSfJwxnvm4z

```
Server: nginx/1.10.3 (Ubuntu)
Date: Sat, 24 Feb 2018 09:02:05 GMT
Content-Type: text/html
Content-Length: 653
Last-Modified: Sat, 24 Feb 2018 09:01:53 GMT
Connection: keep-alive
ETag: "5a912a01-28d"
Accept-Ranges: bytes

<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx. DDCTF{cd5ecb712bd0676f838af04cba1accd}</em></p>
</body>
</html>
```

分组 10179. 1 客户端 分组, 1 服务器 分组, 1 turn(s). 点击选择.

Entire conversation (974 bytes)

显示和保存数据为 ASCII

查找: 滤掉此流 打印 Save as... 返回 Close Help

第四题



```
#!/usr/bin/env python
import sys
import json
from Crypto.Cipher import AES
from Crypto import Random

def get_padding(rawstr):
    remainder = len(rawstr) % 16
    if remainder != 0:
        return '\x00' * (16 - remainder)
    return ''

def aes_encrypt(key, plaintext):
    plaintext += get_padding(plaintext)
    aes = AES.new(key, AES.MODE_ECB)
    cipher_text = aes.encrypt(plaintext).encode('hex')
    return cipher_text

def generate_hello(key, name, flag):
    message = "Connection for mission: {}, your mission's flag is: {}".format(name, flag)
    return aes_encrypt(key, message)

def get_input():
    return raw_input()

def print_output(message):
    print(message)
    sys.stdout.flush()

def handle():
    print_output("Please enter mission key:")
    mission_key = get_input().rstrip()

    print_output("Please enter your Agent ID to secure communications:")
    agentid = get_input().rstrip()
    rnd = Random.new()
    session_key = rnd.read(16)

    flag = '<secret>'
    print_output(generate_hello(session_key, agentid, flag))
    while True:
        print_output("Please send some messages to be encrypted, 'quit' to exit:")
        msg = get_input().rstrip()
        if msg == 'quit':
            print_output("Bye!")
            break
        enc = aes_encrypt(session_key, msg)
        print_output(enc)

if __name__ == "__main__":
    handle()
```

这题算是纯crypto，采用的是MODE_ECB的AES加密。

题目大概的意思就是明文由agentid和flag组成，随意输入 agentid后明文会被随机产生的key进行AES加密，由于每一次远程到116.85.48.103时key都是随机产生的，所以我们只能通过后面的加密尝试来猜测出明文

首先AES了解一下

ECB加密是分组进行加密的，解密也是分组解密。分组与分组之间的明文产生的密文互相独立，且由于算法的缘故，相同的明文分组在相同的密钥加密下会产生相同的密文
加解密流程如下图所示



而我们要做的事是通过这些个分组且明文加密固定密文的特性猜出flag的每一位来
题中以16字节为一组，我们举例也拿16字节为一组举例

首先我们假设xxxx是我们可控的输入，一般情况下的加密会是这样的



但是如果我们将xxxx为十五个固定的字符如十五个A
则加密过程会变成这样：



现在我们记录下此时的HEX_1，在与密文进行比较，如果一样就记下此时F的值，如此一个一个循环推测

以下是解题代码

```
#!/usr/bin/env python
# coding=utf-8
import time
import socket
import string

agentid = ""
message = ""
flag = ""

for i in range(45):
    agentid += "1"

def returnmsg(data):
    """发送与接收"""
    s.send(data)
    time.sleep(0.3)
    msg = s.recv(1024)
    return msg

while True:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("116.85.48.103", 5002))

    print returnmsg("2acba569d223cf7d6e48dee88378288a\n") # 发送题目的mission key
    cipertext = returnmsg(agentid + "\n") # 得到密文
    print cipertext

    for i in string.lowercase + string.digits + string.uppercase + "{}":
        message = "Connection for mission: {}, your mission's flag is: {}".format(agentid, flag + i)
        info = returnmsg(message + "\n")
        if info.split("\n")[0] in cipertext: # 如果尝试加密的内容和第一次密文相同,则记下i
            print info
            flag += i
            print "find key code: " + i
            break

    print message
    s.shutdown(2)
    s.close()
    if agentid == "": # 每一次猜到keycode后,agentid都要减一位
        break
    else:
        agentid = agentid[:-1]
    if i == "}":
        break
```

转载于:<https://www.cnblogs.com/semishigure/p/8930257.html>