Write Up HackToday 2019

itsmine

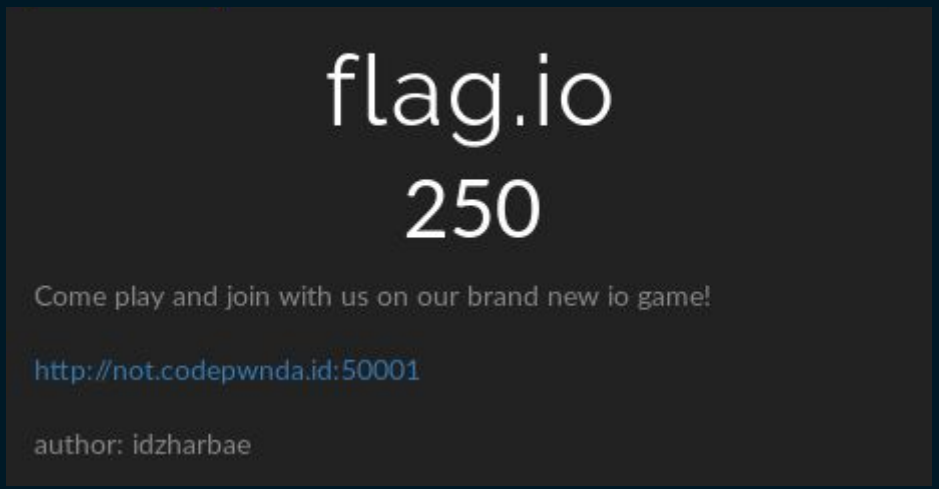
anggota : Achmad Zaenuri Dahlan Putra

misc - sanity check (10 Pts)

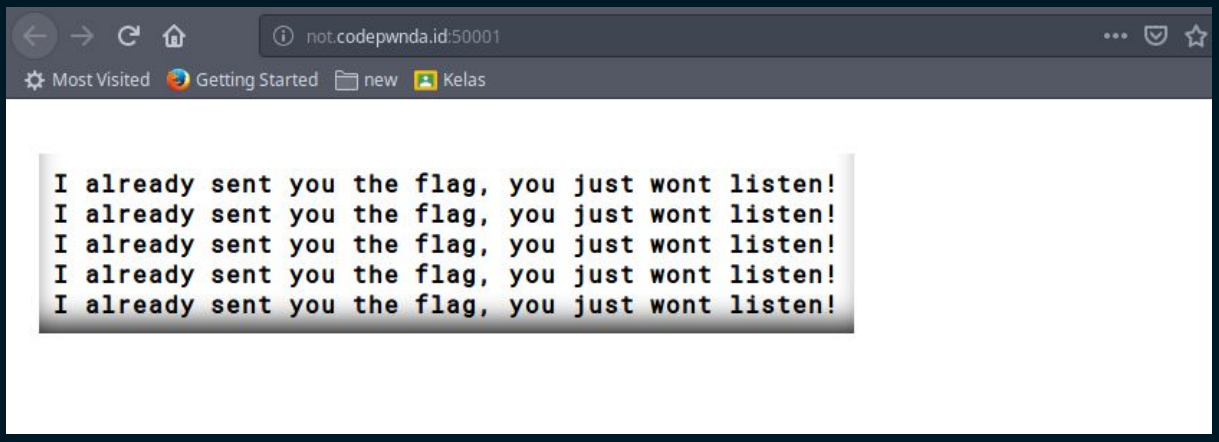
Disini kita cukup mengklik soal dan langsung terdapat flag pada bagian description.

FLAG : hacktoday{sanity_check}

web - flag.io (250 Pts)



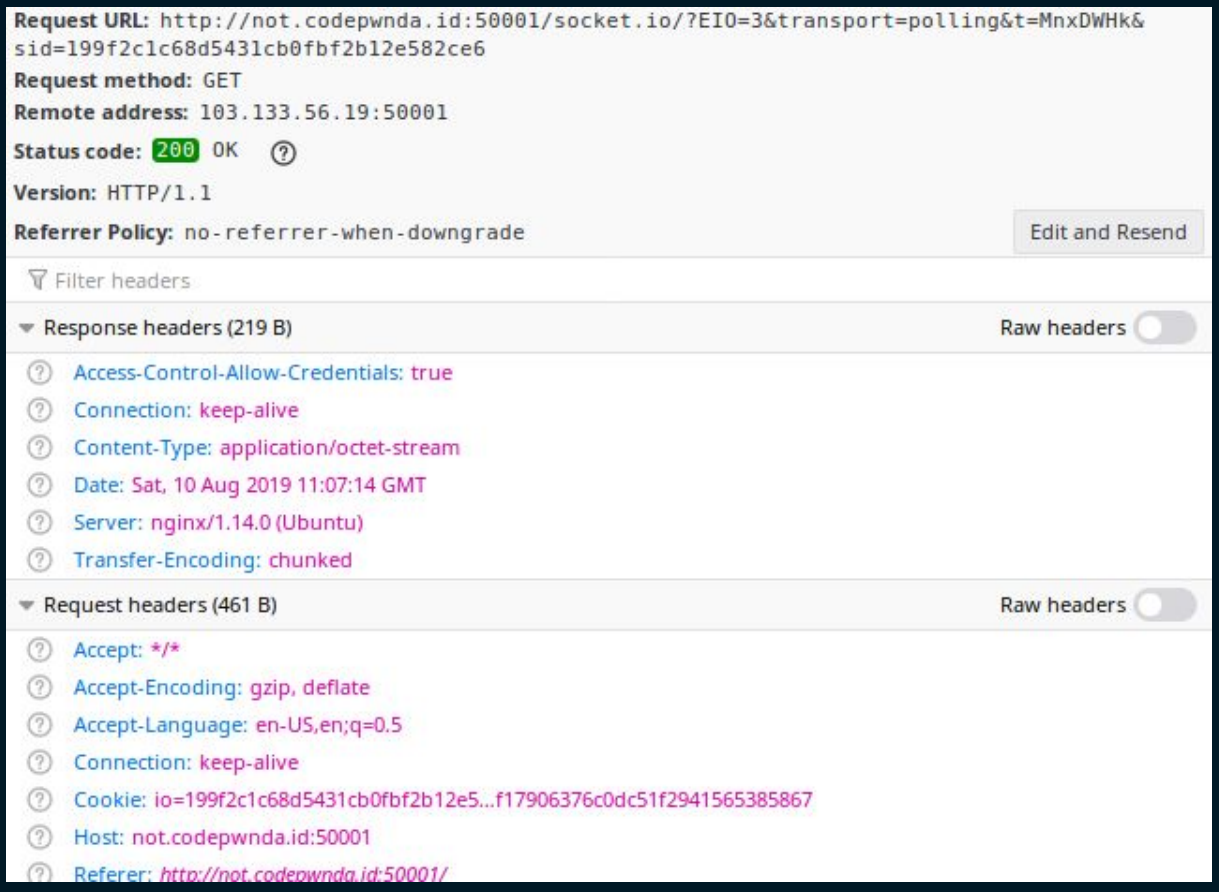
Ketika saya membuka link tersebut maka akan tampil sebagai berikut.



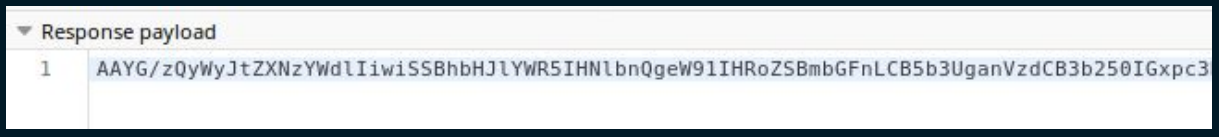
Kemudian saya coba untuk melihat source codenya

```
<html>
    <head>
        <title> Flag IO </title>
        <link href='https://fonts.googleapis.com/css?family=Roboto+Mono' rel='stylesheet'>
        <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
        <link rel="stylesheet" type="text/css" href="/static/css/styles.css">
    </head>
    <body>
        <div class="row">

            <div id="messageBox" class="message_holder">
            </div>

        </div>
        <script src="/static/js/jquery-1.12.4.min.js"></script>
        <script src="/static/js/socket.io-1.7.3.min.js"></script>
        <script>
            var socket = io.connect('http://' + document.domain + ':' + location.port, {
    'sync disconnect on unload': true });
            socket.on('connect', function(){
                console.log('Connected to server.');
              });
            socket.on('message', function(msg){
                $( 'div.message_holder' ).append('<div><b style="color: #000">'+msg+'</div>' )
            });
        </script>
    </body>
</html>
```

Pertama saya sempat berpikir bahwa kita diminta untuk melakukan listening pada server kita lalu melakukan connect menggunakan web tersebut,namun sebelum saya mencobanya saya coba untuk melihat network terlebih dahulu pada firefox untuk mengetahui request dan response yang dilakukan oleh page tersebut.



Dan ternyata terdapat banyak request yang dilakukan oleh page tersebut,sesuai dengan message **I already sent you the flag, you just wont listen!** dan ketika saya lihat responsenya ternyata dalam bentuk base64encode



Response :
AAYG/zQyWyJtZXNzYWdlIiwiSSBhbHJlYWR5IHNlbnQgeW91IHRoZSBmbGFnLCB5b3UganVzdCB3b250IGxpc3RlbiEiXQAGCP80MlsiU1VQQS1TSUtSRVQtRkxBR0dHIiwiaGFja3RvZGF5e0FzX3lvdV9IdW1hbnNfc2F5LF9JbV9hbGxfZWFyc30iXQ==

lalu saya decode dan ketemulah flagnya.

```
noob  kosong  hacktoday  $  echo -n "AAYG/zQyWyJtZXNzYWdlIiwiSSBhbHJlYWR
5IHNlbnQgeW91IHRoZSBmbGFnLCB5b3UganVzdCB3b250IGxpc3RlbiEiXQAGCP80MlsiU1VQQS1T
SUtSRVQtRkxBR0dHIiwiaGFja3RvZGF5e0FzX3lvdV9IdW1hbnNfc2F5LF9JbV9hbGxfZWFyc30iX
Q==" | base64 -d
42["message","I already sent you the flag, you just wont listen!"]42["SUP
A-SIKRET-FLAGGG","hacktoday{As you Humans say, Im all ears}"] noob  kosong
```

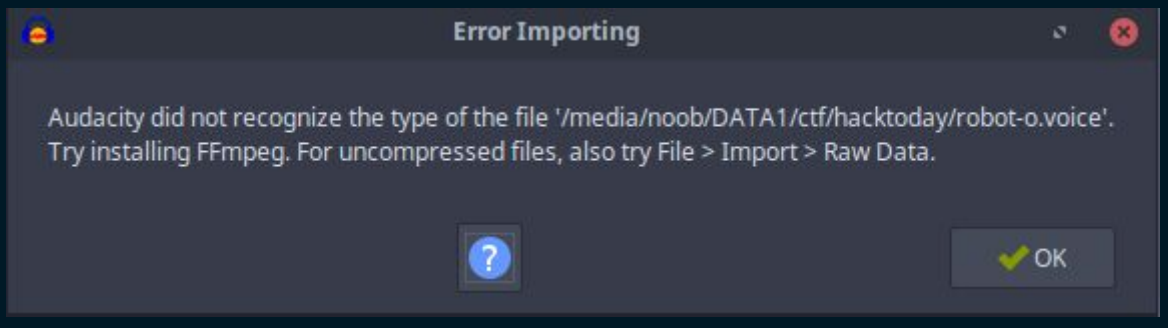
FLAG : hacktoday{As_you_Humans_say,_Im_all_ears}

forensic - robot-o (250 Pts)

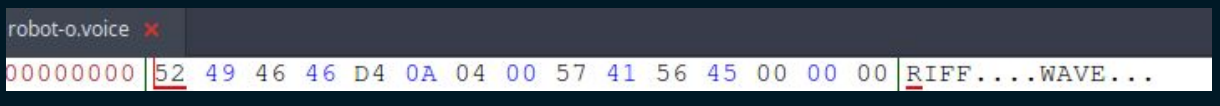
Disini kita diberikan sebuah file dengan nama robot-o.voice,kemudian saya coba cek file tersebut dengan command file pada linux.

```
noob  kosong  hacktoday  $  file robot-o.voice
robot-o.voice: RIFF (little-endian) data, WAVE audio
```

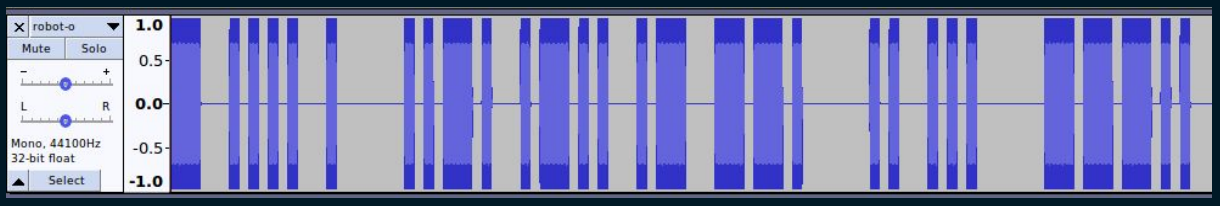
Dan ternyata merupakan file .wav ,jadi selanjutnya saya coba buka menggunakan audacity.



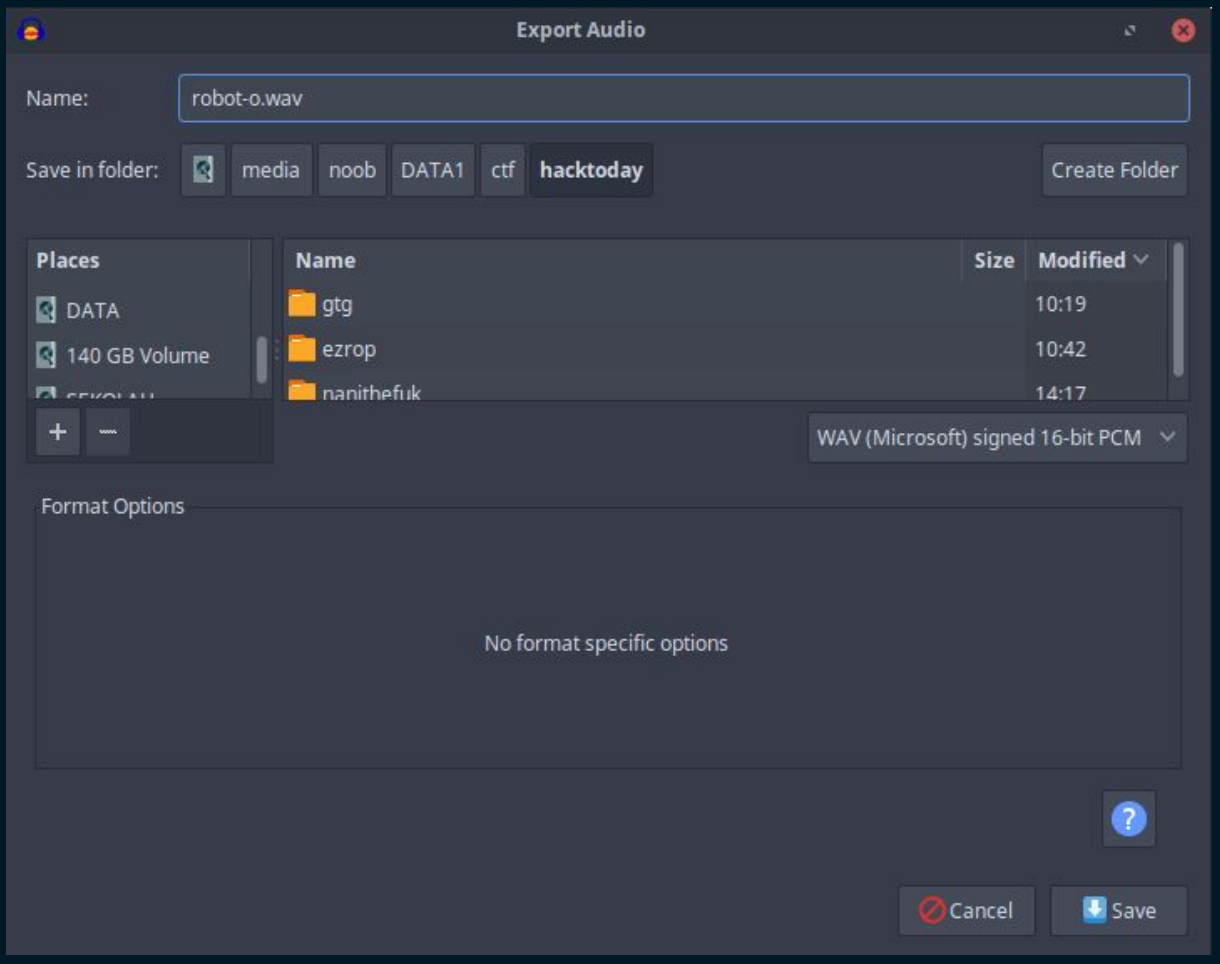
Ternyata terdapat error saat membuka file tersebut,selanjutnya saya coba cek headernya.



Dan ternyata sudah sesuai signaturenya , jadi kemudian saya coba untuk mengikuti notifikasi error saat membuka file tersebut pada audacity.



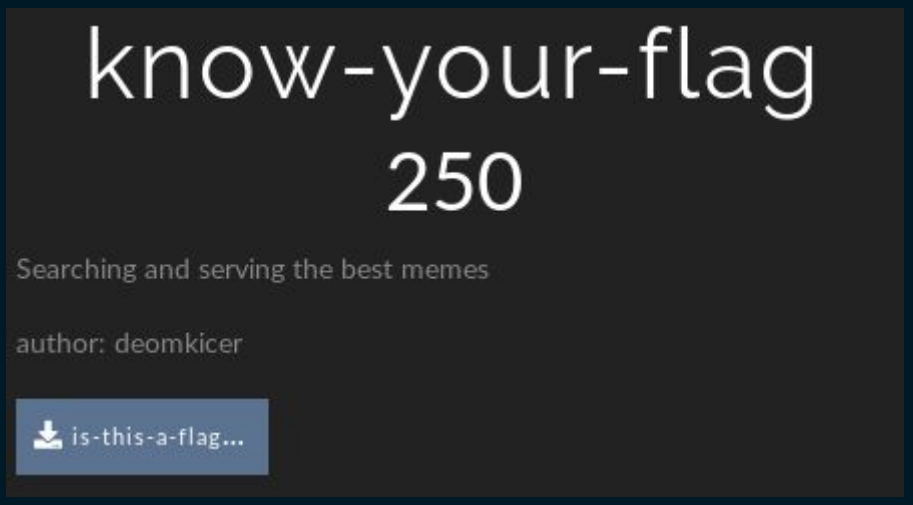
Kalau dilihat dari bentuk gelombang dan suaranya sepertinya ini adalah code morse,jadi saya selanjutnya menyimpan file tersebut sebagai wav lalu mencari morse audio translator.



dan berikut outputnya THE FLAG IS 8AE8CC93E223D5F957CE8B078D2020E7

FLAG : hacktoday{8AE8CC93E223D5F957CE8B078D2020E7}

forensic - know-your-flag (250 Pts)



Disini kita diberikan sebuah file jpg dengan nama is-this-a-flag.
Pertama saya coba buka file tersebut

Dan ternyata tidak terdapat apa apa,kemudian saya melakukan exiftool,binwalk dan tidak membuahkan hasil juga,akhirnya saya coba membuka file image tersebut menggunakan bless(hex editor).
Ternyata pada bagian bawah file tersebut terdapat sebuah string passphare dan juga valuenya.

is-this-a-flag.jpg

```
00095b5f  FB D6 FA 59 6B F3 8F F9 32 EE C1 EF FA 7F 85  ...Yk...2......
00095b6e  14 FA 29 9D 7C EF B7 F5 FD 5F FA 5A FF 00 FF  ..).|...._.Z...
00095b7d  D9 70 61 00 00 73 73 00 00 70 68 00 00 72 61  .pa..ss..ph..ra
00095b8c  00 00 73 65 00 00 3D 39 00 00 38 37 00 00 31  ..se..=9..87..1
00095b9b  32 00 00 33 36 00 00 35 34 00 00 68 6F 00 00  2..36..54..ho..
00095baa  68 6F 00 00 68 6F 00 00                       ho..ho..
```

yang mana jika ditulis menjadi sebagai berikut:
passphrase=987123654hohoho
Karena terdapat clue yaitu sebuah passphare maka langkah selanjutnya saya coba untuk melakukan extract pada file tersebut menggunakan steghide.

```
noob  kosong  hacktoday  $  steghide extract -sf is-this-a-flag.jpg
Enter passphrase:
wrote extracted data to "patrick.jpg".
```

Berhasil lalu kemudian saya analisa file patrick.jpg menggunakan exiftool.

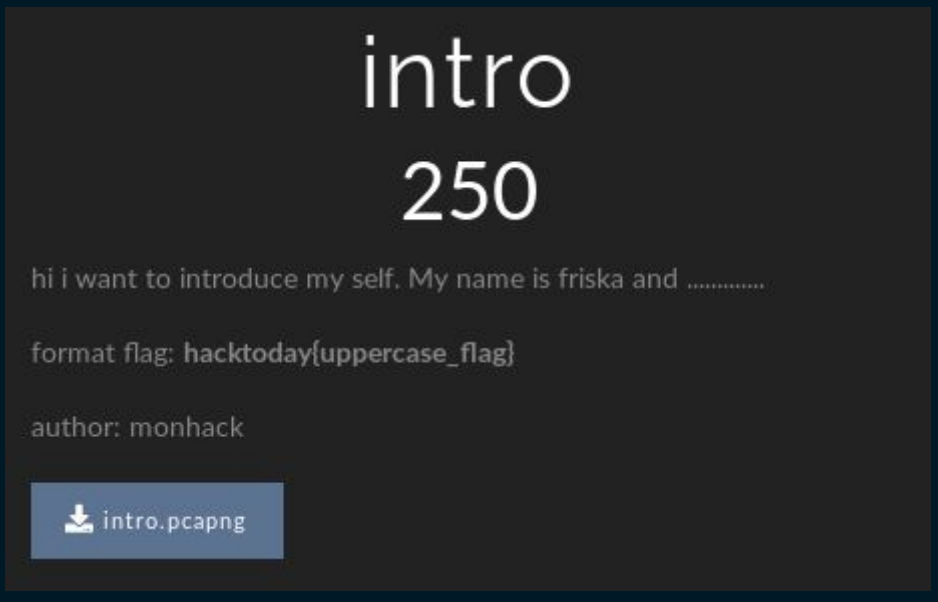
```
noob  kosong  hacktoday  $  exiftool patrick.jpg
ExifTool Version Number         : 10.10
File Name                       : patrick.jpg
Directory                       : .
File Size                       : 5.1 kB
File Modification Date/Time     : 2019:08:10 18:32:38+07:00
File Access Date/Time           : 2019:08:10 18:32:38+07:00
File Inode Change Date/Time     : 2019:08:10 18:32:38+07:00
File Permissions                : rwxrwxrwx
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Comment                         : JJ2XG5BANNUWIZDJNZTS4ICIMVZGKJ3TEB4W65LSEBT
GYYLHHIQGQYLDNN2G6ZDBPF5V6NDMNRPWQNDJNRPV6NLUGM4WQ2LEMVPTCZJSG5RWKM35
Image Width                     : 200
Image Height                    : 148
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
```

Terdapat sesuatu yang menarik,yaitu pada commentnya seperti sebuah string hasil dari base32encode,jadi saya langsung melakukan base32decode pada string tersebut.

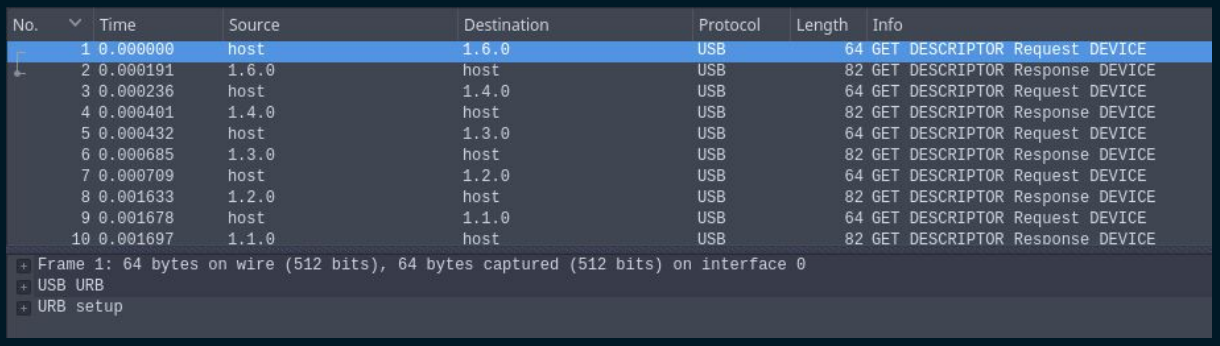
```
noob  kosong  hacktoday  $  echo -n "JJ2XG5BANNUWIZDJNZTS4ICIMVZGKJ3TEB4
W65LSEBTGYYLHHIQGQYLDNN2G6ZDBPF5V6NDMNRPWQNDJNRPV6NLUGM4WQ2LEMVPTCZJSG5RWKM35
" | base32 -d
Just kidding. Here's your flag: hacktoday{_4ll_h4il__5t39hide_1e27ce3} noob
```

FLAG : hacktoday{_4ll_h4il__5t39hide_1e27ce3}

forensic - intro (250 Pts)



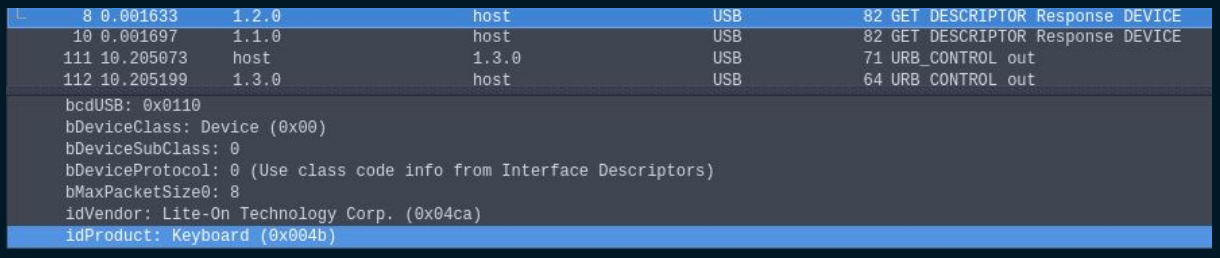
Diberikan sebuah file intro.pcapng kemudian saya langsung membukanya menggunakan wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	host	1.6.0	USB	64	GET DESCRIPTOR Request DEVICE
2	0.000191	1.6.0	host	USB	82	GET DESCRIPTOR Response DEVICE
3	0.000236	host	1.4.0	USB	64	GET DESCRIPTOR Request DEVICE
4	0.000401	1.4.0	host	USB	82	GET DESCRIPTOR Response DEVICE
5	0.000432	host	1.3.0	USB	64	GET DESCRIPTOR Request DEVICE
6	0.000685	1.3.0	host	USB	82	GET DESCRIPTOR Response DEVICE
7	0.000709	host	1.2.0	USB	64	GET DESCRIPTOR Request DEVICE
8	0.001633	1.2.0	host	USB	82	GET DESCRIPTOR Response DEVICE
9	0.001678	host	1.1.0	USB	64	GET DESCRIPTOR Request DEVICE
10	0.001697	1.1.0	host	USB	82	GET DESCRIPTOR Response DEVICE

```
+ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface 0
+ USB URB
+ URB setup
```

Ternyata disini semua traffic menggunakan protocol USB , kemudian saya coba mengurutkan berdasarkan info lalu melihat device apa saja yang digunakan pada traffic ini.

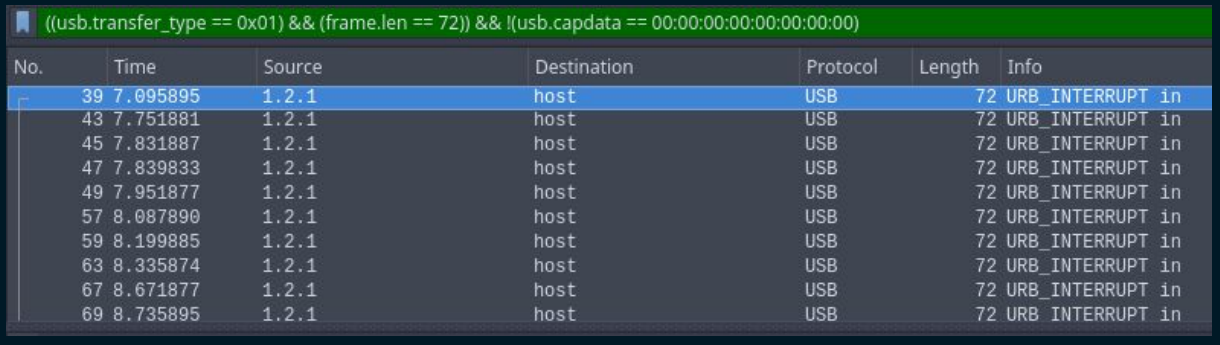


8	0.001633	1.2.0	host	USB	82	GET DESCRIPTOR Response DEVICE
10	0.001697	1.1.0	host	USB	82	GET DESCRIPTOR Response DEVICE
111	10.205073	host	1.3.0	USB	71	URB_CONTROL out
112	10.205199	1.3.0	host	USB	64	URB CONTROL out

```
bcdUSB: 0x0110
bDeviceClass: Device (0x00)
bDeviceSubClass: 0
bDeviceProtocol: 0 (Use class code info from Interface Descriptors)
bMaxPacketSize0: 8
idVendor: Lite-On Technology Corp. (0x04ca)
idProduct: Keyboard (0x004b)
```

Disini terdapat device yang menarik,yaitu keyboard,lalu saya mencoba mencari referensi mengenai bagaimana mengetahui keystroke yang terjadi pada file pcapng tersebut.
Berikut adalah salah satu writeup yang menjadi referensi saya dalam mengerjakan soal ini
https://medium.com/@ali.bawazeeer/kaizen-ctf-2018-reverse-engineer-usb-keystrok-from-pcap-file-2412351679f4

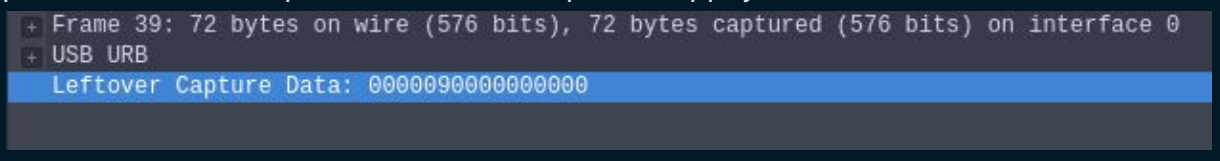
Pertama memfilter traffic hanya untuk traffic input keyboard dengan filter berikut.
((usb.transfer_type == 0x01) && (frame.len == 72)) && !(usb.capdata == 00:00:00:00:00:00:00:00)



((usb.transfer_type == 0x01) && (frame.len == 72)) && !(usb.capdata == 00:00:00:00:00:00:00:00)

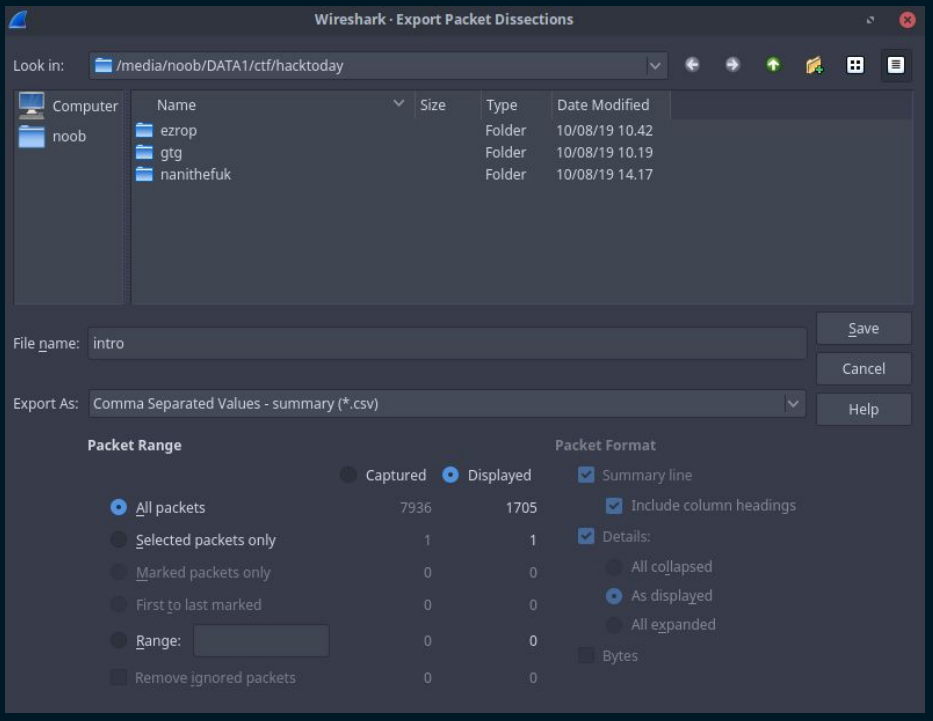
No.	Time	Source	Destination	Protocol	Length	Info
39	7.095895	1.2.1	host	USB	72	URB_INTERRUPT in
43	7.751881	1.2.1	host	USB	72	URB_INTERRUPT in
45	7.831887	1.2.1	host	USB	72	URB_INTERRUPT in
47	7.839833	1.2.1	host	USB	72	URB_INTERRUPT in
49	7.951877	1.2.1	host	USB	72	URB_INTERRUPT in
57	8.087890	1.2.1	host	USB	72	URB_INTERRUPT in
59	8.199885	1.2.1	host	USB	72	URB_INTERRUPT in
63	8.335874	1.2.1	host	USB	72	URB_INTERRUPT in
67	8.671877	1.2.1	host	USB	72	URB_INTERRUPT in
69	8.735895	1.2.1	host	USB	72	URB_INTERRUPT in

kemudian menambah kolom Leftover Capture Data dengan cara klik kanan pada leftover capture data lalu pilih apply as column



```
+ Frame 39: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
+ USB URB
  Leftover Capture Data: 0000090000000000
```

Setelah itu lakukan Export Packet Dissections as CSV agar kita bisa melakukan select pada kolom Leftover Capture Data nantinya.



Kemudian kita ambil data Leftover Capture Data dari intro (csv file)

```
noob  kosong  hacktoday  $  cat intro | cut -d "," -f 7 | cut -d "\"" -f
2 | grep -vE "Leftover Capture Data" > output.txt
noob  kosong  hacktoday  $  head -n 5 output.txt
0000090000000000
0000150000000000
0000150c00000000
00000c0000000000
0000160000000000
```

Setelah data terkumpul selanjutnya kita melakukan mapping dari data data tersebut ke key yang ada pada keyboard dengan script berikut.

```
= File name : cob.py ================================================

newmap = {
 2: "?",
 4: "a",
 5: "b",
 6: "c",
 7: "d",
 8: "e",
 9: "f",
 10: "g",
 11: "h",
 12: "i",
 13: "j",
 14: "k",
```

```
 15: "l",
 16: "m",
 17: "n",
 18: "o",
 19: "p",
 20: "q",
 21: "r",
 22: "s",
 23: "t",
 24: "u",
 25: "v",
 26: "w",
 27: "x",
 28: "y",
 29: "z",
 30: "1",
 31: "2",
 32: "3",
 33: "4",
 34: "5",
 35: "6",
 36: "7",
 37: "8",
 38: "9",
 39: "0",
 40: "Enter",
 41: "esc",
 42: "del",
 43: "tab",
 44: " ",
 45: "-",
 47: "[",
 48: "]",
 55: ".",
 56: "/",
 57: "CapsLock",
 79: "RightArrow",
 80: "LetfArrow"
 }

myKeys = open('output.txt')
i = 1
plain=""
for line in myKeys:
     bytesArray = bytearray.fromhex(line.strip())
 #print "Line Number: " + str(i)
```

```
      for byte in bytesArray:
      if byte != 0:
            keyVal = int(byte)

            if keyVal in newmap:
                  plain+=newmap[keyVal]
            else:
                  print "No map found for this value: " + str(keyVal)

      i+=1
print plain
```

==========

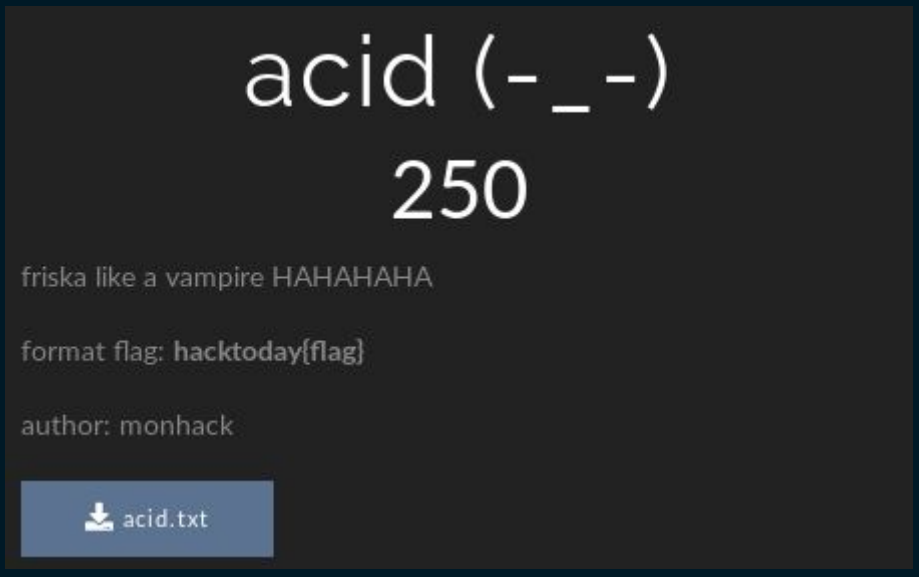
Berikut outputnya :

```
No map found for this value: 51
No map found for this value: 82
No map found for this value: 82
No map found for this value: 82
No map found for this value: 82
No map found for this value: 82
No map found for this value: 82
No map found for this value: 81
No map found for this value: 81
No map found for this value: 81
No map found for this value: 81
No map found for this value: 81
frriiska iiss thhee fast sseectiondeldeldelion ooff the cssaardas a
hungarian folk ddaanccee or of moosst ooff liszt hunngggaariiaan
rhapsodies which takkeedeldeldeldelttaakkee their frdelorm tthhis
ddaanccee the fisdeldelriskkaa is ggeenerraally either turbulleent
or jubilantt  inn  tone griff holland toggeehteerrdeldeldeldeltheerr
with ed bodelrown ffoounddeeedd tthhe businesss  inn
22009delde109del09 bbaaassseed onn  a prriinciplllee
ideloogfgffdeldeldelff ddeelivveerinngg ffeelll  deldeldeleell good
fooodd mmaasdelddeee  fromm  frrees delh quality aanndd
rreesponbility deldeldeldeldeldeldelsibluyydeldely
sourcceedeldelcceed ingredients both founddeers arree
insseeddeldeliddeer 4422 unndddeerr  4422 alumnnii tthhee  company
cydelurrentllyy operraates fldelour bbrrraaanncceehedeldeldelhheeess
nneeeaaarrr  high disseentydeldeldeldeldeldelensity officcee
bbuuildingdeldeldeldelddiinnggs whiitthdeldeldeldeliitth 70 ppeerr
cent oofff  iitts rreevveenedelues cominngg from luncchhttiimmee
trraaddee i ssaaw him wwaaldeldellkinnhh deldeldelhh deldelg around
thee  bbaackyyaard likkee somethinnggs troublinngg hhiimdelm fllaagg
iiss deldeldeldel iiss i-l3arn-us8-c4ptudeldeldelptudelur3 ii
```

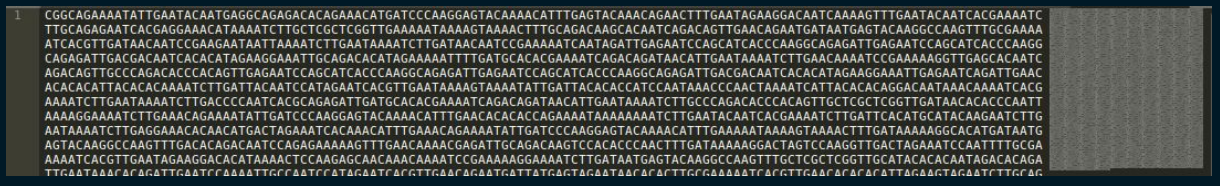
ccaalled hhiim in addelddelnndd when iioi deldel aassked whats goinnggg on hheee just ssaaidd ccaan i ggoo out ffooorrr a whillee i kondeldelwdelnnoow hheee juusstt deldeldeldeldeljust trryyinngg ttooo chhaannggggee tthhe suubbject then aggaain mmaaybbeee hhee juusstt meedeldeldelnneeeded soe deldelmmee frrees airr ttoo clleeeaar idelhhiiiss mddelind ssoo i ssasiidd yyees tthhee endeldelnest deldeldeldelxxtt ddaay i ssaaw hhiim wwaashhiinngg my nneeighbours ccaarr annddd whenn hheee ccaame hhoommee i aasskedd hhiimm why hheee woulldddd ddoo thhaaattdeldeldeldelhhaatdeldelthdeldeldeltadelhhaaattt hheee juusstt ssaaiiddd hhee told mmeee ttooo soo i told hhiim to ttaaakke a bodeldelaathndel anndd do tthhiissdeldelooss deldeldeldeldelhhiiiss hhoommeewedelorkk hdelwhheenn he wwaass donneee i told him thhaat hhee dindeldnt hhaave to wwaashh tthh deldelee ccaar bbeeccaaussee it ssaaxdeldeldeldelwwwaaasss nnooott hiiss rreesponsibbiilitiedeldeliieeess heLetfArrowLetfArrowLetfArrowLetfArrowLetfArrowLetfArrowRightArrowRightArrowRightArrowRightArrowRightArrow kkjjdeljdeldeljuusstt nnood
Flag terdapat pada string fllaagg iiss deldeldeldel iiss i-l3arn-us8-c4ptudeldeldelptudelur3 yang jika di translasikan ke bentuk aslinya menjadi flag is i-l3arn-us8-c4ptur3 .

FLAG : hacktoday{I-L3ARN-US8-C4PTUR3}

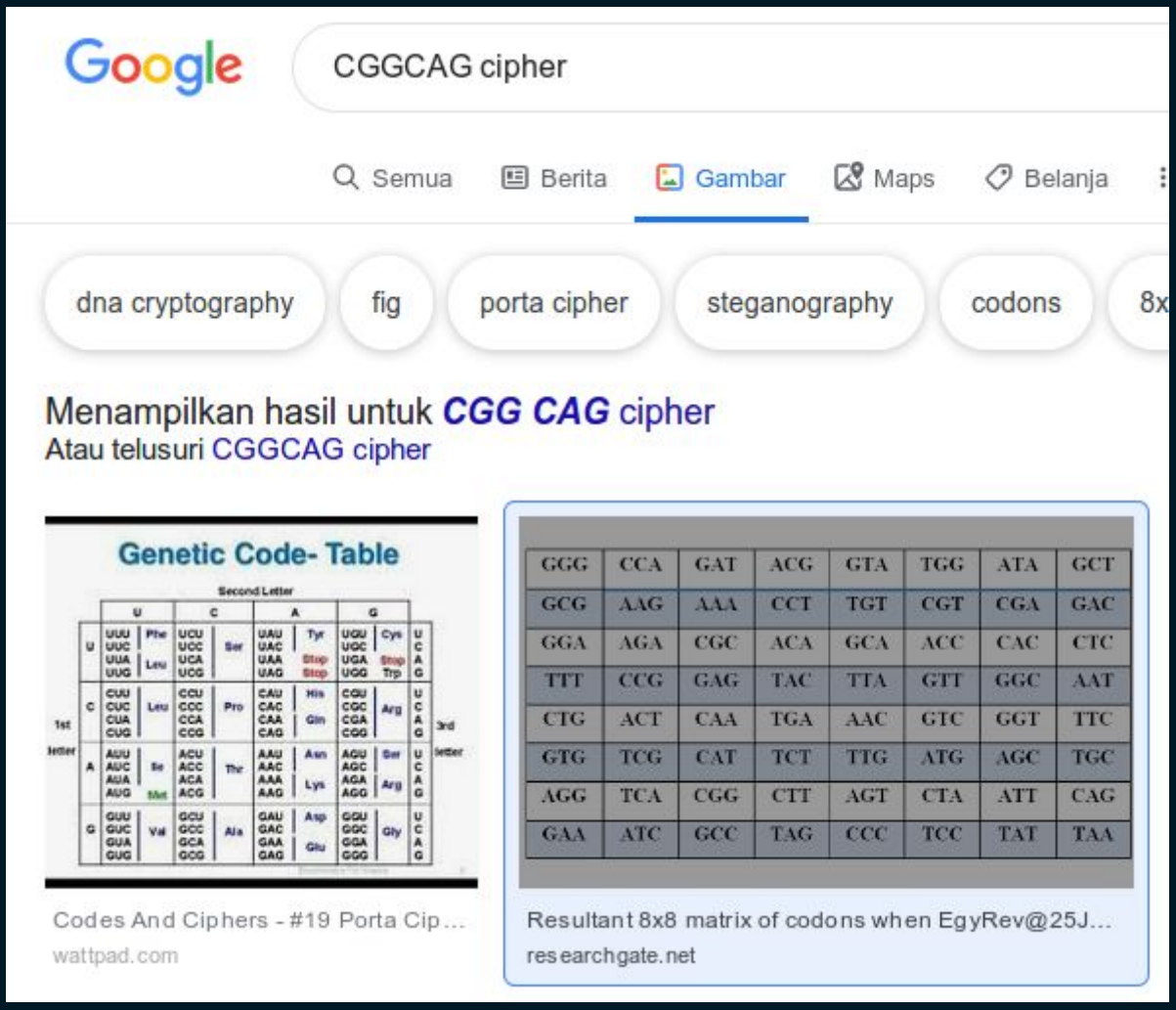
crypto - acid (-_-) (250 Pts)



Diberikan sebuah file acid.txt yang isinya sebagai berikut.



Saya baru kali ini melihat ciphertext seperti ini,jadi saya coba melakukan searching pada beberapa bagian dari text tersebut.



Dan ternyata cipher tersebut dibuat menggunakan table resultan 8x8 matrix of codons. Selanjutny saya coba mencari referensi mengeni codons cipher/DNA Codons cipher dan berikut saya menemukan detail 64 character dari masing masing kolom pada table tersebut.

DNA CODE

Codon	English	Codon	English	Codon	English	Codon	English
AAA	a	CAA	q	GAA	G	TAA	W
AAC	b	CAC	r	GAC	H	TAC	X
AAG	c	CAG	s	GAG	I	TAG	Y
AAT	d	CAT	t	GAT	J	TAT	Z
ACA	e	CCA	u	GCA	K	TCA	1
ACC	f	CCC	v	GCC	L	TCC	2
ACG	g	CCG	w	GCG	M	TCG	3
ACT	h	CCT	x	GCT	N	TCT	4
AGA	i	CGA	y	GGA	O	TGA	5
AGC	j	CGC	z	GGC	P	TGC	6
AGG	k	CGG	A	GGG	Q	TGG	7
AGT	l	CGT	B	GGT	R	TGT	8
ATA	m	CTA	C	GTA	S	TTA	9
ATC	n	CTC	D	GTC	T	TTC	0
ATG	o	CTG	E	GTG	U	TTG	space
ATT	p	CTT	F	GTT	V	TTT	. (period)

Selanjutnya tinggal membuat solver.

```
== File Name : codon.py ========================================
list_char = {
    'AAA' : 'a',
    'AAC' : 'b',
    'AAG' : 'c',
    'AAT' : 'd',
    'ACA' : 'e',
    'ACC' : 'f',
    'ACG' : 'g',
    'ACT' : 'h',
    'AGA' : 'i',
    'AGC' : 'j',
    'AGG' : 'k',
    'AGT' : 'l',
    'ATA' : 'm',
    'ATC' : 'n',
    'ATG' : 'o',
    'ATT' : 'p',
    'CAA' : 'q',
    'CAC' : 'r',
    'CAG' : 's',
    'CAT' : 't',
    'CCA' : 'u',
```

```
    'CCC' : 'v',
    'CCG' : 'w',
    'CCT' : 'x',
    'CGA' : 'y',
    'CGC' : 'z',
    'CGG' : 'A',
    'CGT' : 'B',
    'CTA' : 'C',
    'CTC' : 'D',
    'CTG' : 'E',
    'CTT' : 'F',
    'GAA' : 'G',
    'GAC' : 'H',
    'GAG' : 'I',
    'GAT' : 'J',
    'GCA' : 'K',
    'GCC' : 'L',
    'GCG' : 'M',
    'GCT' : 'N',
    'GGA' : 'O',
    'GGC' : 'P',
    'GGG' : 'Q',
    'GGT' : 'R',
    'GTA' : 'S',
    'GTC' : 'T',
    'GTG' : 'U',
    'GTT' : 'V',
    'TAA' : 'W',
    'TAC' : 'X',
    'TAG' : 'Y',
    'TAT' : 'Z',
    'TCA' : '1',
    'TCC' : '2',
    'TCG' : '3',
    'TCT' : '4',
    'TGA' : '5',
    'TGC' : '6',
    'TGG' : '7',
    'TGT' : '8',
    'TTA' : '9',
    'TTC' : '0',
    'TTG' : ' ',
    'TTT' : '.'
    }

f = open('acid.txt').read().strip()
```

```
flag = ""
for x in range(0,len(f),3):
    code = f[x:x+3]
    flag+=list_char[code]
print flag
```

==========

Berikut outputnya

Asam deoksiribonukleat lebih dikenal dengan singkatan DNA adalah sejenis biomolekul yang menyimpan dan menyandi instruksi instruksi genetika setiap organisme dan banyak jenis virus instruksi instruksi genetika ini berperan penting dalam pertumbuhanperkembangan dan fungsi organisme dan virus DNA merupakan asam nukleat bersamaan dengan protein dan karbohidrat asam nukleat adalah makromolekul esensial bagi seluruh makhluk hidup yang diketahuiKebanyakan molekul DNA terdiri dari dua unting biopolimer yang berpilin satu sama lainnya membentuk heliks gandaDua unting DNA ini dikenal sebagai polinukleotida karena keduanya terdiri dari satuan satuan molekul yang disebut nukleotida tiap tiap nukleotida terdiri atas salah satu jenis basa nitrogen gula monosakarida yang disebut deoksiribosadan gugus fosfat nukleotida nukelotida ini kemudian tersambung dalam satu rantai ikatan kovalen antara gula satu nukleotida dengan fosfat nukelotida lainnyaHasilnya adalah rantai punggung gula fosfat yang berselang seling Menurut kaidah pasangan basa ikatan hidrogen mengikat basa basa dari kedua unting polinukleotida membentuk DNA unting ganda Dua unting DNA bersifat anti paralel yang berarti bahwa keduanya berpasangan secara berlawanan Pada setiap gugus gula terikat salah satu dari empat jenis nukleobasa Urutan urutan empat nukleobasa di sepanjang rantai punggung DNA inilah yang menyimpan kode informasi biologis Melalui proses biokimia yang disebut transkripsi unting DNA digunakan sebagai templat untuk membuat unting RNA Unting RNA ini kemudian ditranslasikan untuk menentukan urutan asam amino protein yang dibangun Struktur kimia DNA yang ada membuatnya sangat cocok untuk menyimpan informasi biologis setiap makhluk hidup Rantai punggung DNA resisten terhadap pembelahan kimia dan kedua dua unting dalam struktur unting ganda DNA menyimpan informasi biologis yang sama Karenanya informasi biologis ini akan direplikasi ketika dua unting DNA dipisahkan Sebagian besar DNA bersifat non kode yang berarti bagian ini tidak berfungsi menyandikan protein Dalam sel DNA tersusun dalam kromosom Semasa pembelahan sel flag is DN4ismybl00d kromosom kromosom ini diduplikasi dalam proses yang disebut replikasi DNA Organisme eukariotik menyimpan kebanyakan DNA nya dalam inti sel dan sebagian kecil sisanya dalam organel seperti mitokondria ataupun kloroplas Sebaliknya organisme prokariotik menyimpan DNA nya hanya dalam sitoplasma Dalam kromosom protein kromatin seperti histon berperan

dalam penyusunan DNA menjadi struktur kompak Struktur kompak inilah yang kemudian berinteraksi antara DNA dengan protein lainnya sehingga membantu kontrol bagian bagian DNA mana sajakah yang dapat ditranskripsikan Para ilmuwan menggunakan DNA sebagai alat molekuler untuk menyingkap teori teori dan hukum hukum fisika seperti misalnya teorema ergodik dan teori elastisitas Sifat sifat materi DNA yang khas membuatnya sangat menarik untuk diteliti bagi ilmuwan dan insinyur yang bekerja di bidang mikrofabrikasi dan nanofabrikasi material Beberapa kemajuan di bidang material ini misalnya origami DNA dan material hibrida berbasis DNA
Dan tara terdapat flag pada string diatas.
FLAG : hacktoday{DN4ismybl00d}

crypto - rsa-goes-skrrraahh (438 Pts)

Diberikan file enc dan rsa-goes-skrrrhh.py , disini saya coba menganalisa file rsa-goes-skrrrhh.py .
Disini intinya untuk p dan q digenerate secara random menggunakan function generateN lalu kita diberikan sebuah hint yang merupakan hasil perkalian dari p+x dengan q+x , berikut potongan kodenya

```
def generateKeys(p, q):
      e = 0x10001
      n = p * q
```

```
    phi = (p-1) * (q-1)
    d = modinv(e, phi)
    h = (p+0x69420) * (q+0x69420)
    return [e, n, h]
```

disini kita diberikan string value dari e n h ,karena kita tahu bahwa h merupakan (p+0x69420) * (q+0x69420) dan n = p*q jadi kita dapat menggunakan z3 untuk mencari value dari p dan q.
Berikut script untu mencari value dari p dan q

```
from z3 import *

x = Real('x')
y = Real('y')
solve(x * y ==
3931057479215986708746504999454647816614616133587799630551156682834832374445396020696347739510077736802054572777146963626825617441034009953891819320478509812486244457181370331899074262930587799069690953005197820710050882842952218995938680206230325498760489136992875076760593999307042295390557802467746722575161301522252369325458753689149767484505516805177851766680209043182161752479968000246655375676524647760198168646825338417822999924695817146431381939101557643082009525845029167895317478958544716653095995283728046691176435790369478143904715072221579162623390516475408174023812106193129905565299384183034607677427,
(x+0x69420) * (y+0x69420) ==
3931057479215986708746504999454647816614616133587799630551156682834832374445396020696347739510077736802054572777146963626825617441034009953891819320478509812486244457181370331899074262930587799069690953005197820710050882842952218995938680206230325498760489136992875076760593999307042295390557802467746776931053048572187847797252282068198564595788076002007898520713515668527316461313547558044098710118442626837912008250075879359585523066510180827832526630849756052633293445318623478999832320790958696194699642596107256651255924849047384853241528176039659410185036114818966403826291209555272283812911955997679965116275)
```

kemudian didapatkan value dari p dan q sebagai berikut

```
noob  kosong  hacktoday  $  python3 /home/noob/reversing/z3/real.py
[x = 565011370609950339366875902288887665017571573052796605438439504444547914
883030358424161991690903199350569039121168952224418605998276081731131326645552
533511053259964663995909521968454415373961939537484082250798529481483418414233
319511734912163258094575814733252917754342440675696048058778605792458191969934
46839,
 y = 695748383784253240906443735328454630617942106138846635830070508790842893
463032915799233814814107792957637135121752150410038775489772743102881175236582
823805891763302434830153702422699454499656626250438066786424943413287789485498
392842620207797352574245395776015686127338948199431709723722093210124000971762
693]
```

Kemudian kita tinggal melakukan rsa decrypt dengan script berikut

```
from Crypto.Util.number import long_to_bytes
import gmpy

def hitung(p,q):
    e = 65537
    c =
35682206022280308942342358335623063320594568312574708624904778423328
89106330357717497935562917168335744490294025013583483080963708070777
14255972039887443854744601219076522717728952522607224786445508552326
56934126084594212628107240004990627184072900379298442070061594300739
23653091573172702861239928443239320542778093734748954368249583256502
20560440433253840809025948013482898644222268927889071894679587967150
10296808164629775416885567796667835855934493093419696807485899353196
28657917542748251463546769215830052399157026963904886704565816634869
97630359553715253350306790371651641243009111022953367484680861418072
0700
    n = p*q
    phi = (p-1)*(q-1)
    d = gmpy.invert(e, phi)
    m = pow(c, d, n)
    return long_to_bytes(m)

print
hitung(565011370609950339366875902288887665017571573052796605438439504444547914
883030358424161991690903199350569039121168952224418605998276081731131326645525
335110532599646399590952196845441537396193953748408225079852948148341841423319
511734912163258094575814733252917754342440675696048058778605792458191969934683
9,695748383784253240906443735328454630617942106138846635830070508790842893463
032915799233814814107792957637135121752150410038775489772743102881175236588238
805891763302434830153702422699454499656626250438066786424943413287789485498392
842620207797352574245395776015686127338948199431709723722093210124000971762693)
```

Berikut outputnya

```
noob  kosong  hacktoday  $  python rsa.py
hacktoday{p_plus_q_is_solved__quick_maths_}~~~~~~~~~~~~~~~~~~~~~~
```

FLAG : hacktoday{p_plus_q_is_solved__quick_maths_}

Thank You ! :)