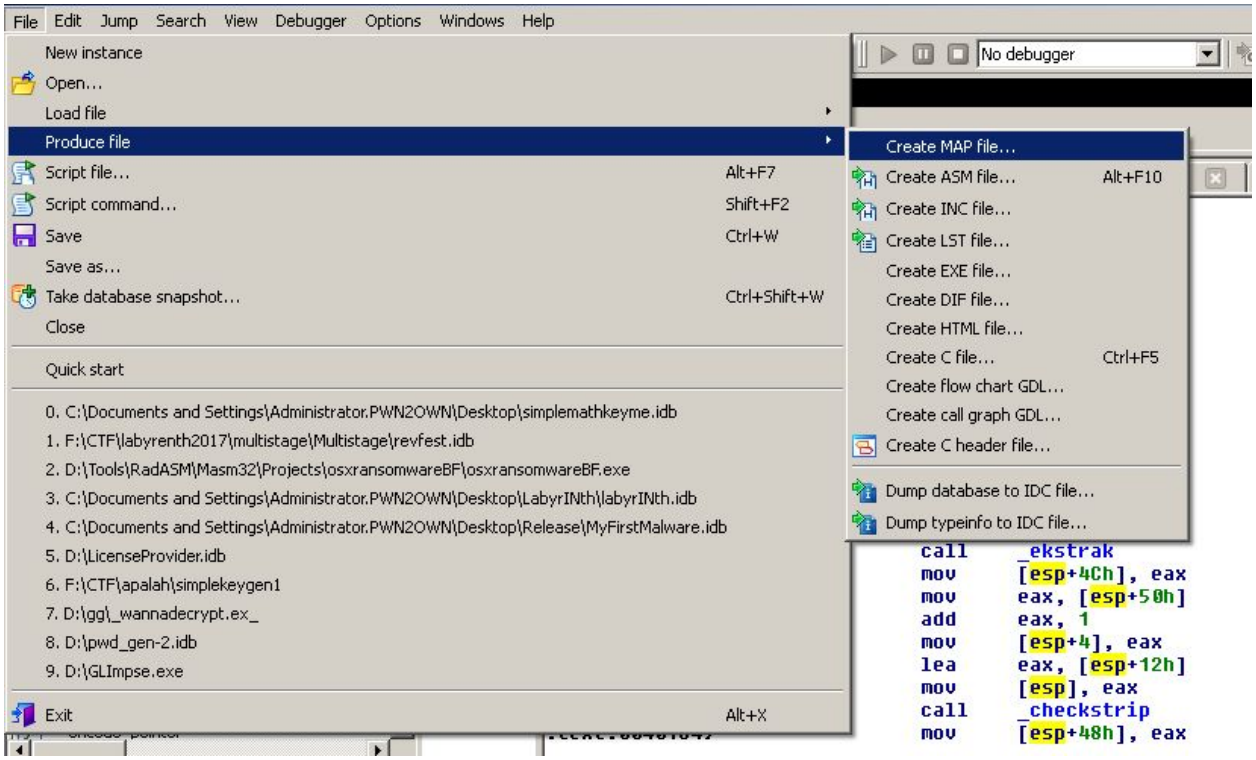
Selamat malam,

Kulgram kali ini kita bahas soal dari mas @darkro5e : simplemathkeyme
Terima kasih buat mas @darkro5e yang udah bikin keyme nya, dan udah bikin versi windows nya.

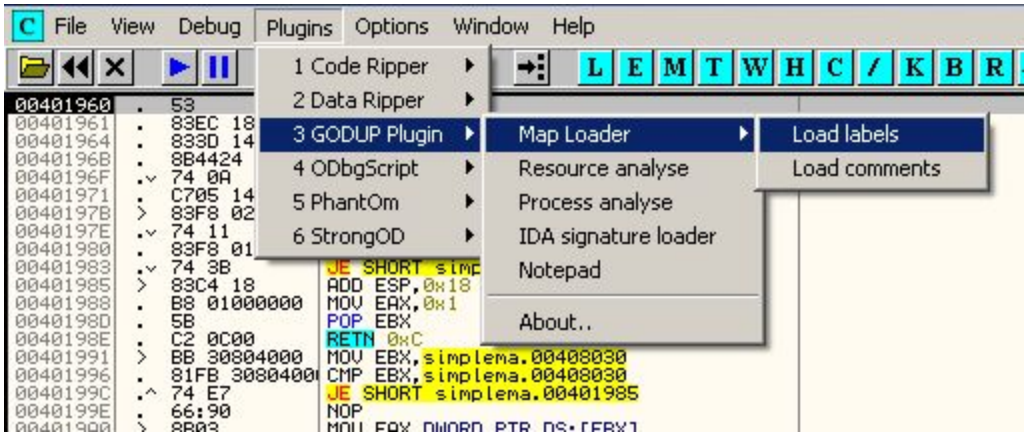
Untuk tool yang dipakai, IDA Pro (maap masih pake bajakan..) sama OllyDBG + GODUPPlugin.

GODUP untuk load file .map yang digenerate lewat IDA

Create map file dari IDA:



Load map via di odbg pake plugin GODUP



Setelah load exe nya di IDA, kita coba decompile fungsi main

```
  v3 = check((int)&v4);
  if ( v3 != panjang((int)&v4) )
          notserial();
  v11 = panjang((int)&v4);
  v10 = checkstrip((int)&v4, 0);
  v9 = ekstrak((int)&v4, 0, v10);
  v8 = checkstrip((int)&v4, v10 + 1);
  v7 = ekstrak((int)&v4, v10 + 1, v8);
  v6 = checkstrip((int)&v4, v8 + 1);
  v5 = ekstrak((int)&v4, v8 + 1, v6);
  if ( !(v7 % v11)
          && !(v9 % v11)
          && !(v5 % v11)
          && v9
          && (v9 < v7) < v5
          && v5 * (v7 * v9 / (v9 * (v7 / v9))) / (v5 * v11) == 1 )
          benar();
  salah();
```

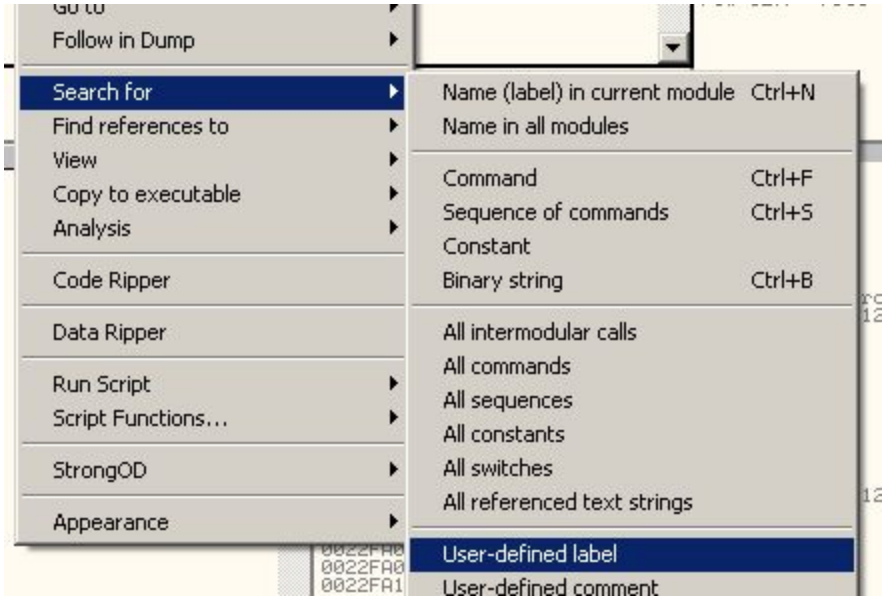
Dari posisi ini lumayan jelas alur programnya ngapain aja.
Karena exe nya ga di-strip, jadi beberapa nama fungsi masih bisa 'dibaca' oleh IDA

Kita akan cek satu per satu fungsi fungsi itu ngapain aja.

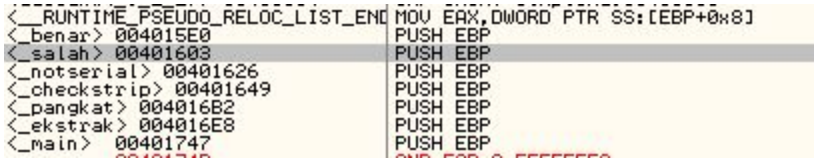
Ada beberapa hal yang ane pakai untuk dijadikan acuan kalo ngerjain crackme :

1. Return value dari sebuah fungsi biasanya ada di eax
2. Ngga perlu terlalu detail 'baca' kode assembly nya. Kadang ane pake metode blackbox. Inputnya apa, outputnya apa.

Let's debug. Load programnya ke odbg
dari odbg, search for user generated label



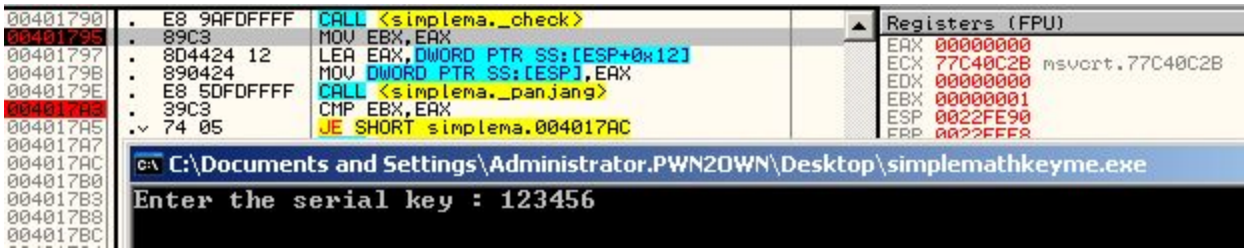
Label yang tadi ada di IDA pro, bisa kita lihat di odbg juga :



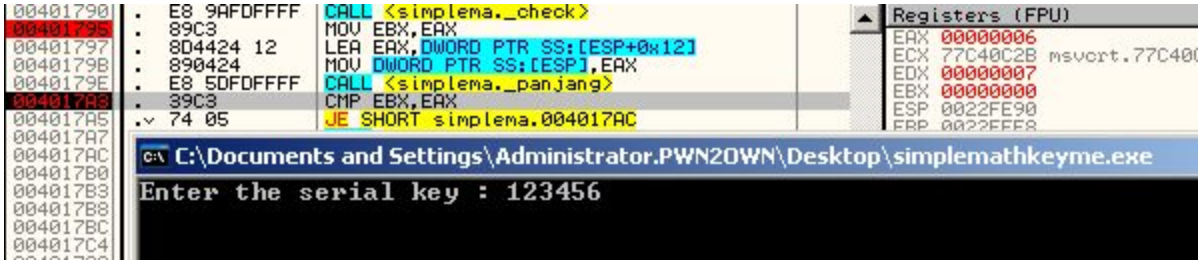
kita bakalan nyari tau, fungsi check dan panjang itu ngapain aja..
caranya, coba beberapa input text dan lihat hasil fungsi check dan fungsi panjang

pasang breakpoint di
00401795 dan **004017A3** (setelah pemanggilan fungsi), jalanin debuggernya (F9).

untuk input text : **123456**
hasil fungsi check : **0** (lihat eax)



hasil fungsi panjang : **6** (lihat eax)



disini jelas kalo fungsi panjang akan menghitung jumlah karakter dari input teks (jelas banget sih dari namanya...)

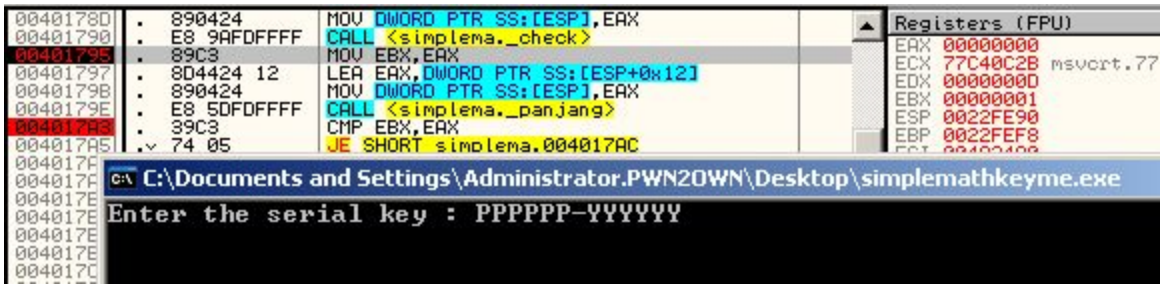
kita bakal pake bantuan hexrays decompiler dari IDA Pro buat analisis fungsi check

```
 for ( i = 0; i <= v2; ++i )
 {
        if ( *(_BYTE *)(i + a1) <= 79 || *(_BYTE *)(i + a1) > 89 )
        {
        if ( *(_BYTE *)(i + a1) != 45 || v2 - 1 <= i || !v5 )
        break;
        ++v4;
        v5 = 0;
        }
        else
        {
        ++v6;
        ++v5;
        }
 }
 if ( v4 == 2 )
        result = v6 + 2;
 else
        result = 0;
 return result;
```

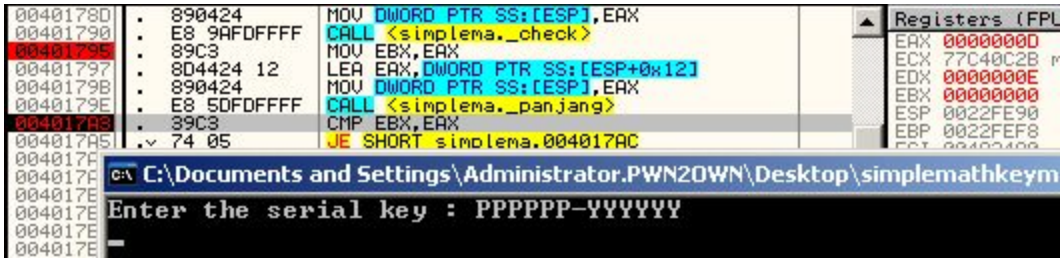
sekilas dibaca, karakter yang akan diproses disini adalah yang berada di rentang **80** sampai **89** (ascii "**P**" sampai "**Y**")
dan ada pengecekan terhadap karakter **45** (ascii "**-**") jika karakter tidak berada di rentang tersebut

kita coba input baru : **PPPPPP-YYYYYY**

hasil fungsi check : **0**



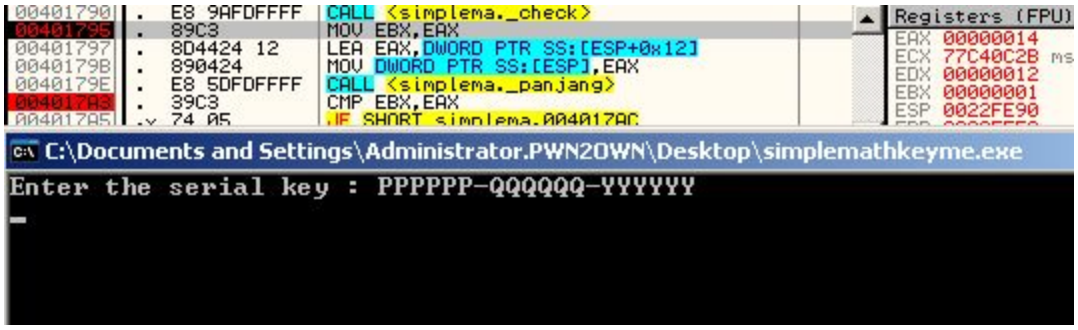
hasil fungsi panjang : **13 (0x0D)**



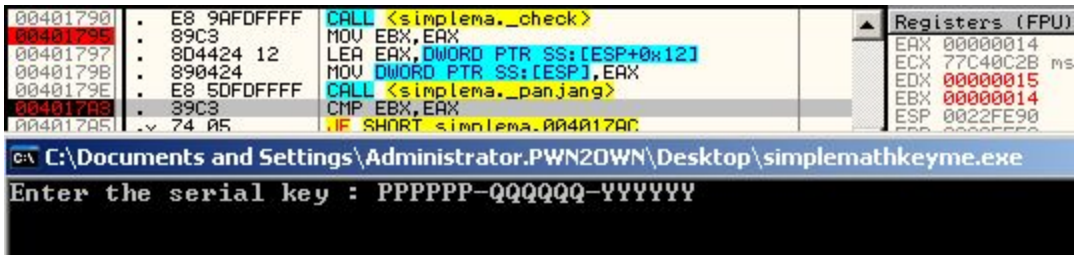
masih salah...

kita coba input baru : **PPPPPP-QQQQQQ-YYYYYY**

hasil fungsi check : **20 (0x14)**



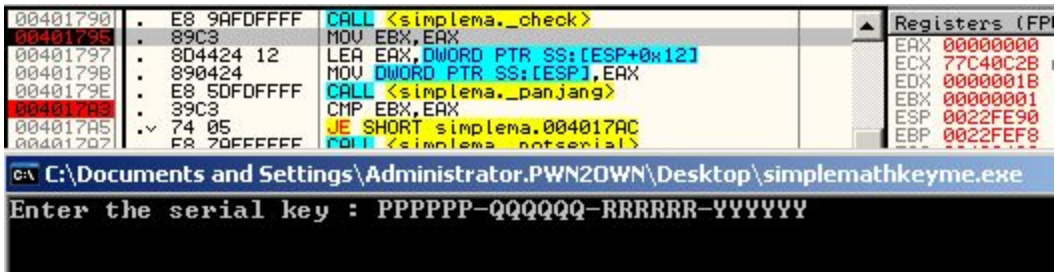
hasil fungsi panjang : **20 (0x14)**



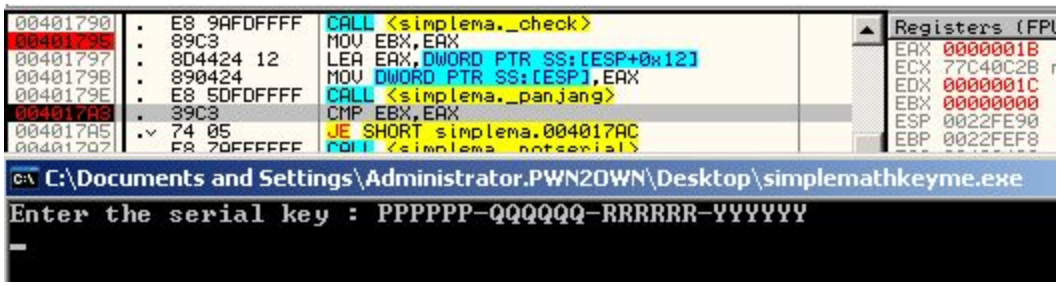
udah bener..

kita coba input baru : **PPPPPP-QQQQQQ-RRRRRR-YYYYYY**

hasil fungsi check : **0**



hasil fungsi panjang : **27 (0x1B)**



salah lagi...

dari 3 percobaan diatas, bisa ane simpulkan kalo format yang diterima oleh programnya adalah :

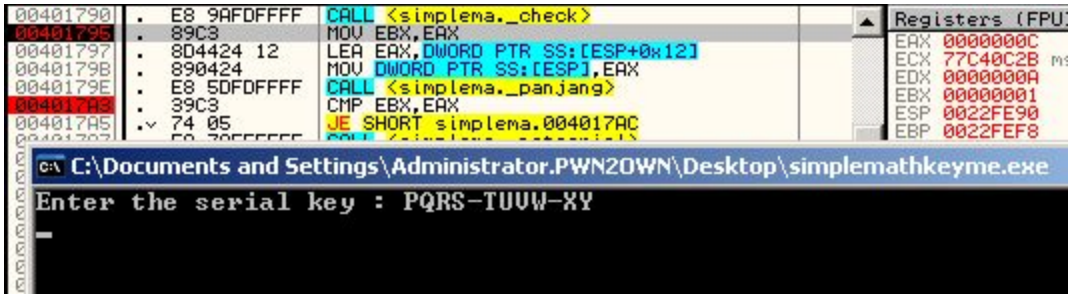
N-N-N

dimana N ada di rentang "**P**" sampai "**Y**", dan terdapat **2** karakter "**-**"

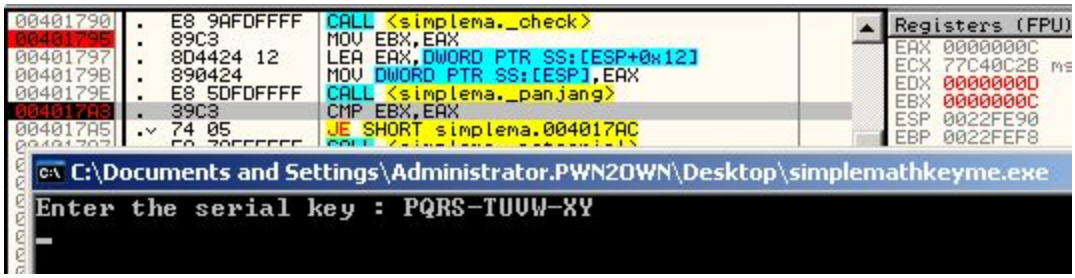
mari verifikasi lagi...

kita coba input baru : **PQRS-TUVW-XY**

hasil fungsi check : **12 (0x0C)**



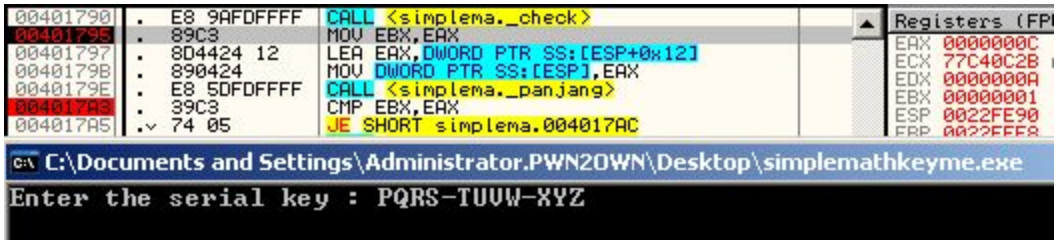
hasil fungsi panjang : **12 (0x0C)**



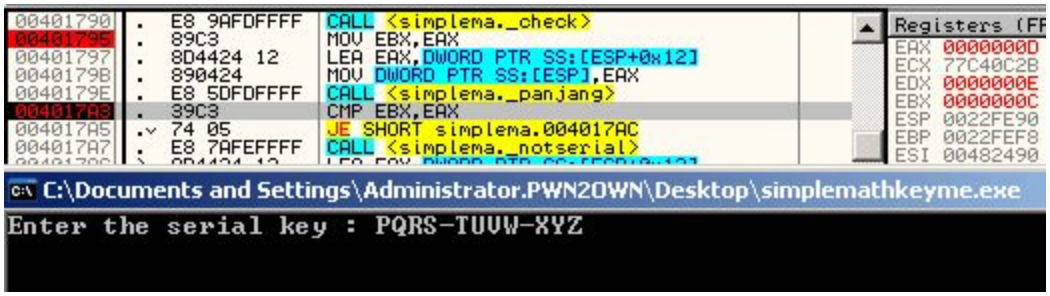
valid

kita coba input baru : **PQRS-TUVW-XYZ**

hasil fungsi check : **12 (0x0C)**



hasil fungsi panjang : **13 (0x0D)**



ga valid ("Z" ga masuk range)

kita coba input baru : **PQRS-TUVW-XY-PPP**

hasil fungsi check : **0**

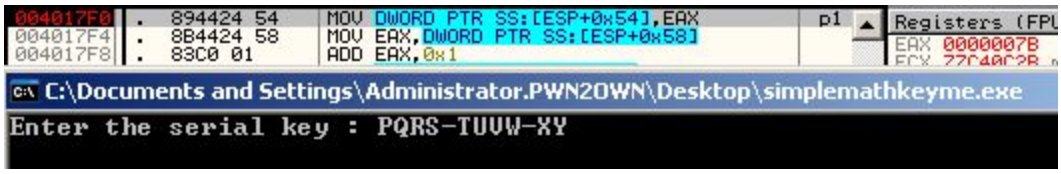
hasil fungsi panjang : **16 (0x10)**

ga valid (ada 3 karakter "-")

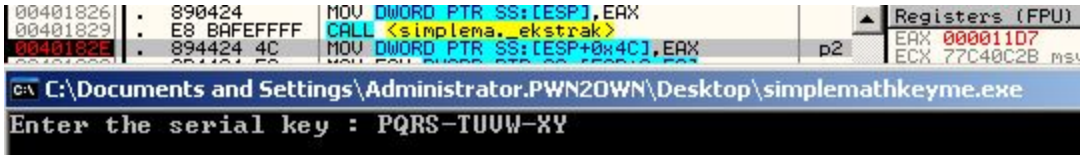
tinggal 2 fungsi yang tersisa : checkstrip dan ekstrak
disable semua breakpoint, pasang breakpoint baru di akhir pemanggilan fungsi ekstrak

004017F0
0040182E
0040186C

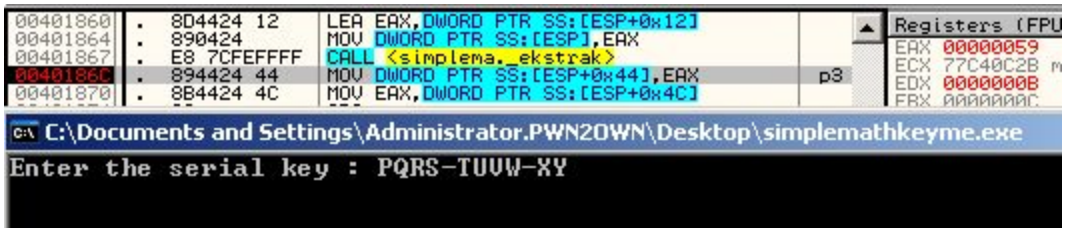
kita coba input baru : **PQRS-TUVW-XY**
hasil ekstrak1 : **123 (0x7B)**



hasil ekstrak2 : **4567 (0x11D7)**



hasil ekstrak3 : **89 (0x59)**



dari hasil ini, bisa disimpulkan kalo fungsi kedua fungsi tersebut adalah fungsi untuk '**mentranslate**' input **text** ke **integer**
dan kalo dibandingkan antara format input dan integer yang dihasilkan, bisa kita simpulkan kalo

fungsi ekstrak, akan mengkonversi input ("**P**" sampai "**Y**") ke angka desimal (**0** sampai **9**)
dimana **"P"=0, "Q"=1, ... , "Y" = 9**

P Q R S T U V W X Y
0 1 2 3 4 5 6 7 8 9

sehingga input **PQRS-TUVW-XY** akan dikonversi ke angka **0123-4567-89**

lanjut..

dari hasil dekompilasi HexRays :
```
  v11 = panjang((int)&v4);
  v9 = ekstrak((int)&v4, 0, v10); ekstrak1
  v7 = ekstrak((int)&v4, v10 + 1, v8); ekstrak2
  v5 = ekstrak((int)&v4, v8 + 1, v6); ekstrak3

  if ( !(v7 % v11)
          && !(v9 % v11)
          && !(v5 % v11)
          && v9
          && (v9 < v7) < v5
          && v5 * (v7 * v9 / (v9 * (v7 / v9))) / (v5 * v11) == 1 )
          benar();
```

kondisi agar inputan teks yang kita input itu benar :

- nilai v7,v9,v5 harus kelipatan v11 (panjan inputan text)
- v9 != 0
- (v9 < v7) < v5
- v5 * (v7 * v9 / (v9 * (v7 / v9))) / (v5 * v11) == 1

nah.. kita selesaikan dulu persamaan

v5 * (v7 * v9 / (v9 * (v7 / v9))) / (v5 * v11) == 1

v5 * (v7 * v9 / (v9 * (v7 / v9))) = v5 * v11

(v7 * v9 / (v9 * (v7 / v9))) = v11

dari rumus persamaan

```
      b
a * ---- = b
      a
```

persamaan :
```
           v9
v7 * -----------
           v7       =  v11
     v9 * ----
           v9
```

disederhanakan menjadi :
```
       v9
v7 * ---- = v11
       v7
```

disederhanakan lagi :

v9 = v11

nah... rumusnya udah dapet,

syarat serial valid :

format : v9-v7-v5

kondisi :

- v9 = v11 = panjang serial

- v7 & v5 kelipatan v11

contoh serial valid :

PX-WR-VT
08-72-64

Panjang serial (v11) : 8
v9 = 08 (v11)
v7 = 72 (kelipatan v11)
v5 = 64 (kelipatan v11)