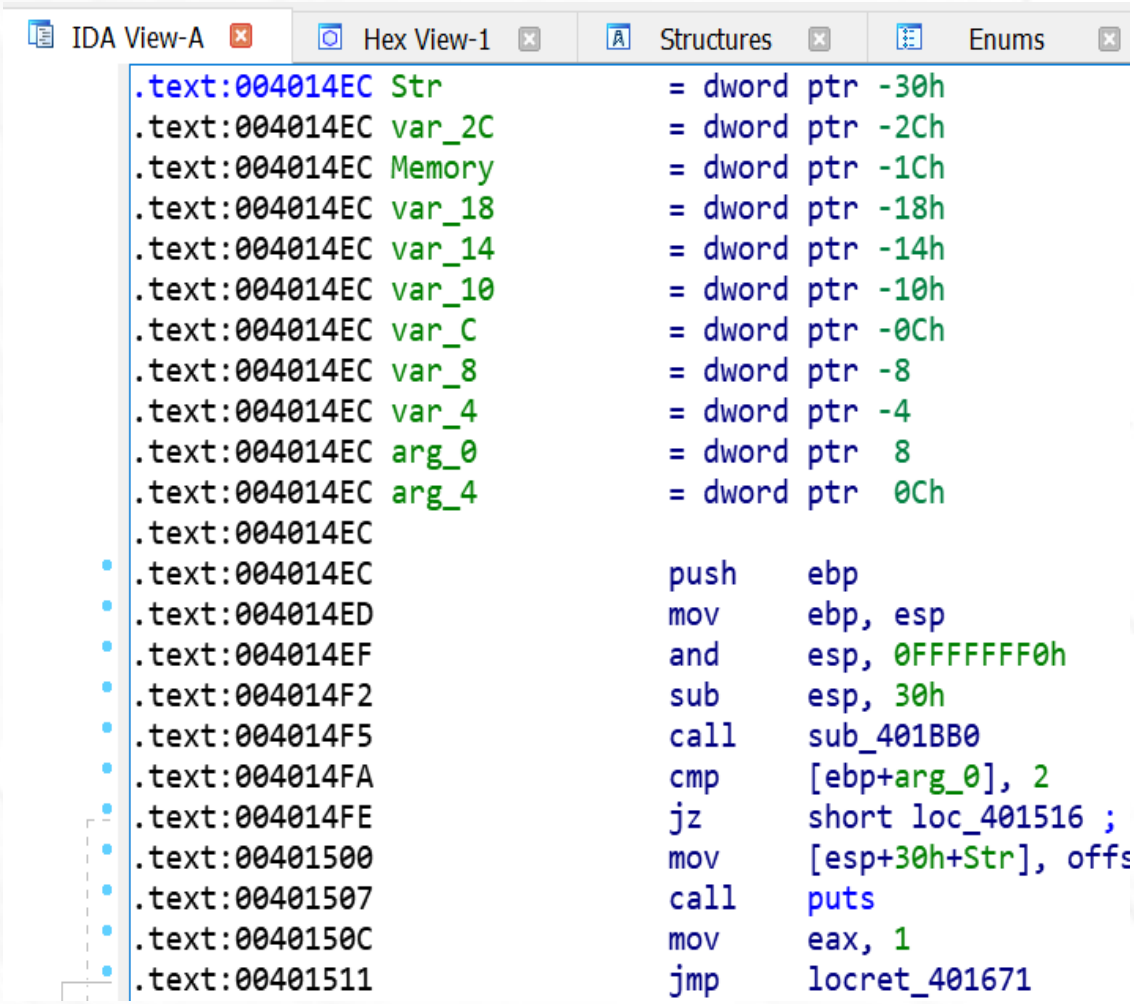
# CTF reverse engineering basics

Pasi Hakkarainen

Jysec 30.8.2019

# Stick to the basics

- Some theory
- Tools
- Hands-on



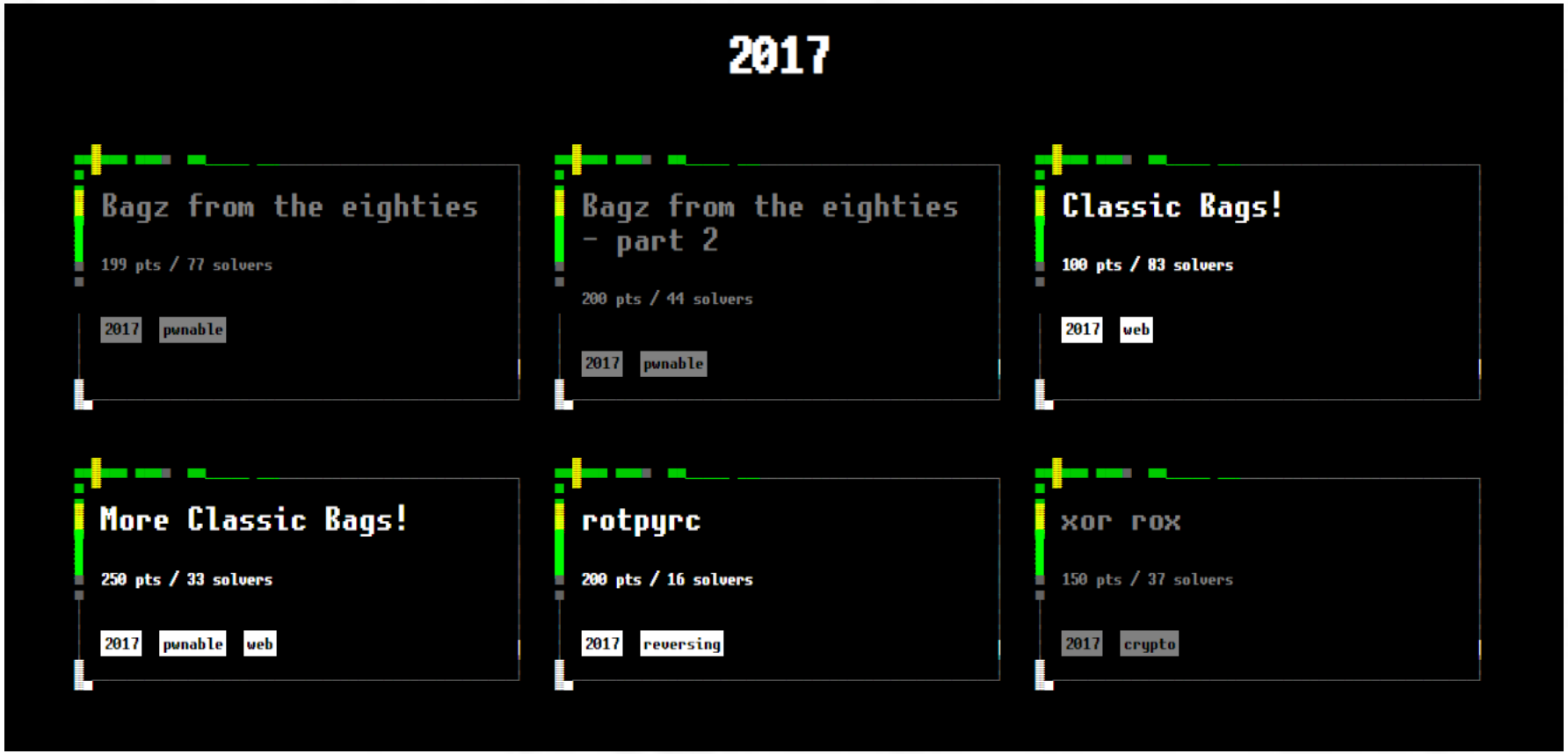

Juho

Reverse

# Pasi Hakkarainen

- Teacher at the moment
- Learned reverse from internet by myself
- Reverse engineering experience about > 1 year
  - Nixu challenge
  - HTB
  - Some email malware
  - Web -injections

# Example


2017
Bagz from the eighties
199 pts / 77 solvers
2017 pwnable
Bagz from the eighties - part 2
200 pts / 44 solvers
2017 pwnable
Classic Bags!
100 pts / 83 solvers
2017 web
More Classic Bags!
250 pts / 33 solvers
2017 pwnable web
rotpyrc
200 pts / 16 solvers
2017 reversing
xor rox
150 pts / 37 solvers
2017 crypto

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

/* whoops. it was from last year anyway. :)

// Set this to 1 only in -dev and -staging
// production!!!!11
#define ENABLE_DEBUG_SHELL 0

#define SO_MANY_BAGZ 8

static char banner[] = {
        #include "banner.h"
};
static char Flag[64];
static char Username[32] = {0, 1};
static char Unprivileged = 1;

struct bag {
        char reporter[32];
        char desc[128];
        int severity;
};

static struct bag Bagz[SO_MANY_BAGZ];
static int Next;

void
menu()
{
        printf("\n"
               "1) Submit a bag\n"
               "2) Show bagz\n"
               "3) ???\n"
               "4) Profit!\n"
#if ENABLE_DEBUG_SHELL
               "5) Debug shell\n"
```

```
text:08048AB5                 public mainloop
text:08048AB5 mainloop        proc near               ; CODE XREF: main
text:08048AB5
text:08048AB5 var_14          = byte ptr -14h
text:08048AB5 var_C           = dword ptr -0Ch
text:08048AB5 var_4           = dword ptr -4
text:08048AB5
text:08048AB5                 push    ebp
text:08048AB6                 mov     ebp, esp
text:08048AB8                 push    ebx
text:08048AB9                 sub     esp, 14h
text:08048ABC                 call    __x86_get_pc_thunk_bx
text:08048AC1                 add     ebx, 253Fh
text:08048AC7                 sub     esp, 8
text:08048ACA                 lea     eax, (aRb - 804B000h)[ebx] ; "rb"
text:08048AD0                 push    eax
text:08048AD1                 lea     eax, (aFlag1Txt - 804B000h)[ebx]
text:08048AD7                 push    eax
text:08048AD8                 call    _fopen
text:08048ADD                 add     esp, 10h
text:08048AE0                 mov     [ebp+var_C], eax
text:08048AE3                 sub     esp, 4
text:08048AE6                 push    [ebp+var_C]
text:08048AE9                 push    3Fh
text:08048AEB                 lea     eax, (Flag - 804B000h)[ebx]
text:08048AF1                 push    eax
text:08048AF2                 call    _fgets
text:08048AF7                 add     esp, 10h
text:08048AFA                 sub     esp, 0Ch
text:08048AFD                 lea     eax, (Flag - 804B000h)[ebx]
```

# Tools

- IDA
  - the world's smartest and most feature-full disassembler,
- DnSpy
  - dnSpy is a debugger and .NET assembly editor.
- Ollydbg
  - OllyDbg is a 32-bit assembler level analysing debugger for Microsoft® Windows®
- Android studio
  - Android Studio provides the fastest tools for building apps on Android device.

# Tools

- Edb-debug
  - edb is a cross platform AArch32/x86/x86-64 debugger.
- file, strings, gdb, objdump, xxd, binwalk
  - Linux
- Apktool, jd-gui
  - Linux

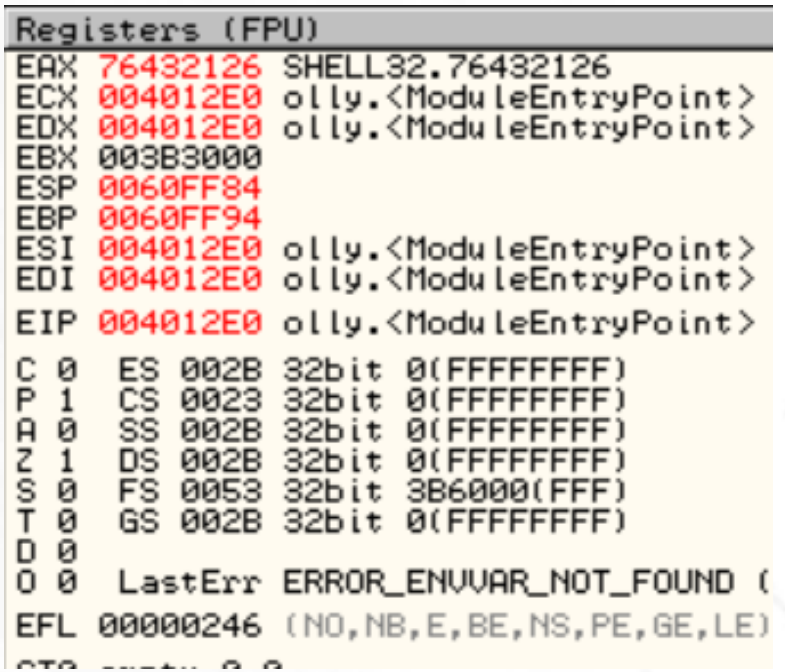
# Skills

**Programming**

- C
- Assembly
  - push ebp
  - cmp eax, 10h
  - jne 0805ff66
- Scripting
  - linux, python, perl etc.

# Skills

- How computer works
  - Process, memory, registery, stack
- Data formats
  - Binary 00010011
  - Hex 10h
  - Hex to Ascii  50 61 48 61 0a
- Buffer overflow
  - Strings and buffer overflow
  - Buffer overflow exploit



# Exercise: Simple reversing

- https://pastebin.com/88DMtBGr
- wget https://pastebin.com/raw/88DMtBGr
- base64 -di 88DMtBGr > simple
- chmod 700 simple
- ./simple
  - Enter passnumber:
- Not working? Is the binary broken?
- wc -c 88DMtBGr
  - 22882

# Basic research

- Identify the file
  - PE, portable executable
  - ELF, Executable and Linkable Format
  - Bits 32/64
  - Little-endian/big-endian
- File, objdump -f, binwalk -B
- Strings
- ./ (you should not run malware)

| cmp dst, src | ZF | CF |
|---|---|---|
| dst = src | 1 | 0 |
| dst < src | 0 | 1 |
| dst > src | 0 | 0 |

# Basic reverse

- Ida64
  - Read the flag
  - Search and calculate "salanumero"
- Edb-debugger
  - Change jump condition je
  - Change register, stack or flag Z
  - Read the value
    - Register
    - stack

# Assembly

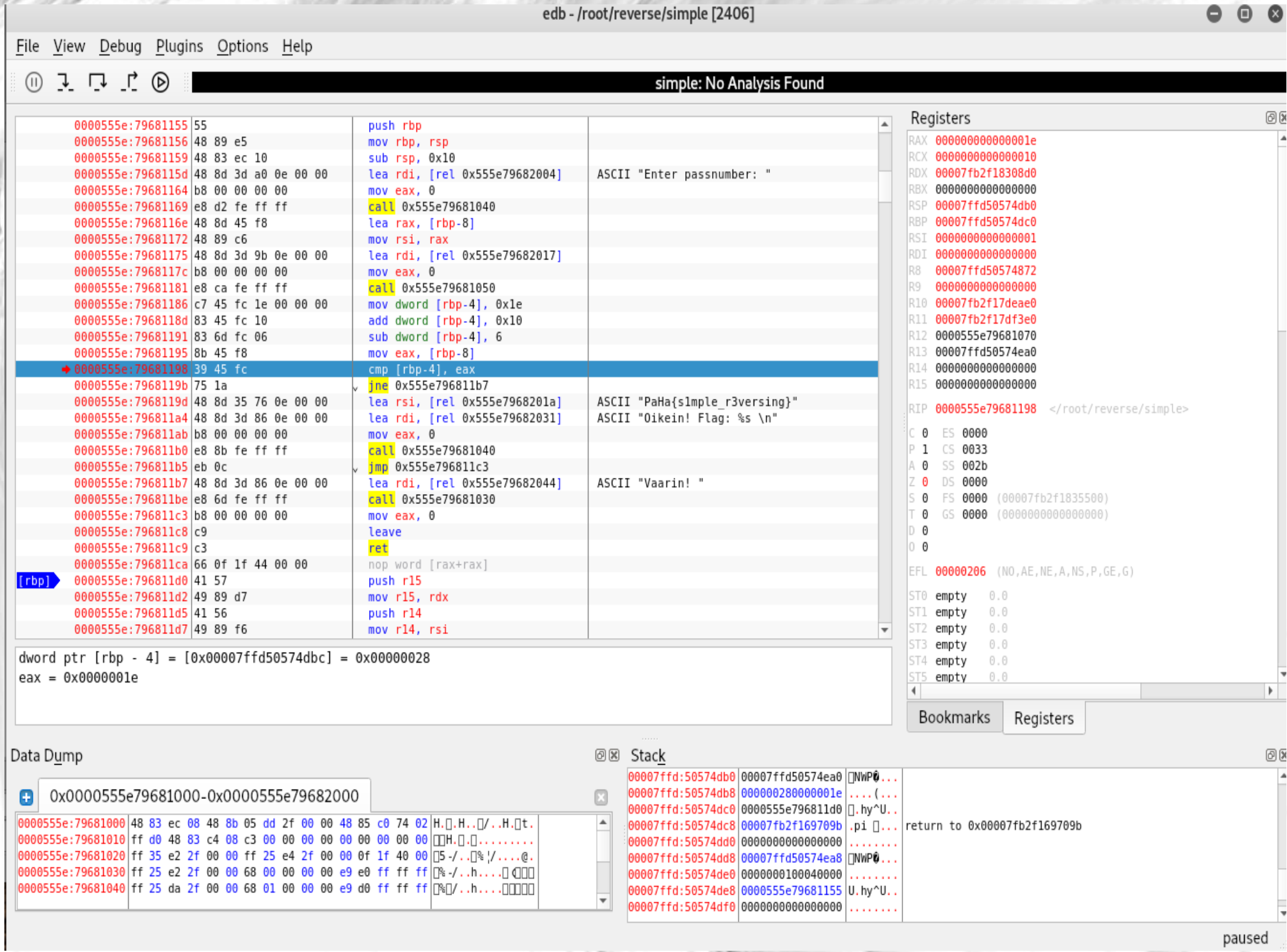
- Assembly Programming Tutorial

```
.text:0000000000001155                  public main
.text:0000000000001155 main             proc near               ; DATA XREF: _start+1D↑o
.text:0000000000001155
.text:0000000000001155 var_8            = dword ptr -8
.text:0000000000001155 var_4            = dword ptr -4
.text:0000000000001155
.text:0000000000001155                  push    rbp
.text:0000000000001156                  mov     rbp, rsp
.text:0000000000001159                  sub     rsp, 10h        ; memory space in stack for variables
.text:000000000000115D                  lea     rdi, format     ; "Enter passnumber: "
.text:0000000000001164                  mov     eax, 0
.text:0000000000001169                  call    _printf         ; print
.text:000000000000116E                  lea     rax, [rbp+var_8]
.text:0000000000001172                  mov     rsi, rax
.text:0000000000001175                  lea     rdi, aD         ; "%d"
.text:000000000000117C                  mov     eax, 0
.text:0000000000001181                  call    ___isoc99_scanf ; read the user input
.text:0000000000001186                  mov     [rbp+var_4], 1Eh
.text:000000000000118D                  add     [rbp+var_4], 10h
.text:0000000000001191                  sub     [rbp+var_4], 6
.text:0000000000001195                  mov     eax, [rbp+var_8]
.text:0000000000001198                  cmp     [rbp+var_4], eax
.text:000000000000119B                  jnz     short loc_11B7
.text:000000000000119D                  lea     rsi, aPahaS1mpleR3ve ; "PaHa{s1mple_r3versing}"
.text:00000000000011A4                  lea     rdi, aOikeinFlagS ; "Oikein! Flag: %s \n"
.text:00000000000011AB                  mov     eax, 0
.text:00000000000011B0                  call    _printf
.text:00000000000011B5                  jmp     short loc_11C3
```

# C

```
/**************************************************
*********PaHa's software*************************
**************************************************/
#include <stdio.h>
# define FLAG "PaHa{s1mple_r3versing}"

int main(void){
   int passu;
   printf("Enter passnumber: ");
   scanf("%d", &passu);
   int sala=30;
   sala=(sala+16);
   sala=(sala-6);
   if(passu==sala){
      printf("Oikein! Flag: %s \n", FLAG);}
   else {
      printf("Vaarin! \n");}
}
```


edb - /root/reverse/simple [2406]
File View Debug Plugins Options Help
simple: No Analysis Found
0000555e:79681155 55 push rbp
0000555e:79681156 48 89 e5 mov rbp, rsp
0000555e:79681159 48 83 ec 10 sub rsp, 0x10
0000555e:7968115d 48 8d 3d a0 0e 00 00 lea rdi, [rel 0x555e79682004] ASCII "Enter passnumber: "
0000555e:79681164 b8 00 00 00 00 mov eax, 0
0000555e:79681169 e8 d2 fe ff ff call 0x555e79681040
0000555e:7968116e 48 8d 45 f8 lea rax, [rbp-8]
0000555e:79681172 48 89 c6 mov rsi, rax
0000555e:79681175 48 8d 3d 9b 0e 00 00 lea rdi, [rel 0x555e79682017]
0000555e:7968117c b8 00 00 00 00 mov eax, 0
0000555e:79681181 e8 ca fe ff ff call 0x555e79681050
0000555e:79681186 c7 45 fc 1e 00 00 00 mov dword [rbp-4], 0x1e
0000555e:7968118d 83 45 fc 10 add dword [rbp-4], 0x10
0000555e:79681191 83 6d fc 06 sub dword [rbp-4], 6
0000555e:79681195 8b 45 f8 mov eax, [rbp-8]
0000555e:79681198 39 45 fc cmp [rbp-4], eax
0000555e:7968119b 75 1a jne 0x555e796811b7
0000555e:7968119d 48 8d 35 76 0e 00 00 lea rsi, [rel 0x555e7968201a] ASCII "PaHa{s1mple_r3versing}"
0000555e:796811a4 48 8d 3d 86 0e 00 00 lea rdi, [rel 0x555e79682031] ASCII "Oikein! Flag: %s \n"
0000555e:796811ab b8 00 00 00 00 mov eax, 0
0000555e:796811b0 e8 8b fe ff ff call 0x555e79681040
0000555e:796811b5 eb 0c jmp 0x555e796811c3
0000555e:796811b7 48 8d 3d 86 0e 00 00 lea rdi, [rel 0x555e79682044] ASCII "Vaarin! "
0000555e:796811be e8 6d fe ff ff call 0x555e79681030
0000555e:796811c3 b8 00 00 00 00 mov eax, 0
0000555e:796811c8 c9 leave
0000555e:796811c9 c3 ret
0000555e:796811ca 66 0f 1f 44 00 00 nop word [rax+rax]
[rbp] 0000555e:796811d0 41 57 push r15
0000555e:796811d2 49 89 d7 mov r15, rdx
0000555e:796811d5 41 56 push r14
0000555e:796811d7 49 89 f6 mov r14, rsi
dword ptr [rbp - 4] = [0x00007ffd50574dbc] = 0x00000028
eax = 0x0000001e
Registers
RAX 000000000000001e
RCX 0000000000000010
RDX 00007fb2f18308d0
RBX 0000000000000000
RSP 00007ffd50574db0
RBP 00007ffd50574dc0
RSI 0000000000000001
RDI 0000000000000000
R8 00007ffd50574872
R9 0000000000000000
R10 00007fb2f17deae0
R11 00007fb2f17df3e0
R12 0000555e79681070
R13 00007ffd50574ea0
R14 0000000000000000
R15 0000000000000000
RIP 0000555e79681198 </root/reverse/simple>
C 0 ES 0000
P 1 CS 0033
A 0 SS 002b
Z 0 DS 0000
S 0 FS 0000 (00007fb2f1835500)
T 0 GS 0000 (0000000000000000)
D 0
O 0
EFL 00000206 (NO,AE,NE,A,NS,P,GE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
Bookmarks
Registers
Data Dump
0x0000555e79681000-0x0000555e79682000
0000555e:79681000 48 83 ec 08 48 8b 05 dd 2f 00 00 48 85 c0 74 02 H.□.H..□/..H.□t.
0000555e:79681010 ff d0 48 83 c4 08 c3 00 00 00 00 00 00 00 00 00 □□H.□.□.........
0000555e:79681020 ff 35 e2 2f 00 00 ff 25 e4 2f 00 00 0f 1f 40 00 □5-/..□%¦/....@.
0000555e:79681030 ff 25 e2 2f 00 00 68 00 00 00 00 e9 e0 ff ff ff □%-/..h....□□□□
0000555e:79681040 ff 25 da 2f 00 00 68 01 00 00 00 e9 d0 ff ff ff □%□/..h....□□□□
Stack
00007ffd:50574db0 00007ffd50574ea0 □NWP□...
00007ffd:50574db8 000000280000001e ....(...
00007ffd:50574dc0 0000555e79681d0 □.hy^U..
00007ffd:50574dc8 00007fb2f169709b .pi □... return to 0x00007fb2f169709b
00007ffd:50574dd0 0000000000000000 ........
00007ffd:50574dd8 00007ffd50574ea8 □NWP□...
00007ffd:50574de0 0000000100040000 ........
00007ffd:50574de8 0000555e79681155 U.hy^U..
00007ffd:50574df0 0000000000000000 ........
paused

# Where to start

- Nixu challenge
- Hackthebox
- Jysec challenge:
- https://pastebin.com/qfYWaDWP
- Wget https://pastebin.com/raw/qfYWaDWP
- Base64 -di qfYWaDWP > target
- Chmod 700 target

XOR EAX, EAX