Hacking with armv7

G.Filosofi 2018

Programmers are only concerned of source code. Hackers are focused on the build products.
Here we use nasm, nm, objdump, gcc, ld, gdb and many other useful commands.
From time to time we work on macOS or GNU/Linux.

- This is our extension convention

file.c	source
file.asm	assembly
file.o	object
file	executable

- Let's create a simple C program from Terminal, ex3.c

```c
#include <stdio.h>
int main(void) {
    printf("Hello, world!\n");
    return 0;
}
```

Build it
$ gcc -O0 -o ex3 ex3.c
Note: For a c++ program we might have had
$ c++ -g -o ex1 ex1.cpp

$ file ex3
ex3: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, for GNU/Linux 3.2.0, BuildID[sha1]=385729f3d52714516479738d92fc6e9a194a0ba0, not stripped

Display assembly
$ objdump -d ex3 | grep main -A8

```
0001043c <main>:
   1043c:   e92d4800    push    {fp, lr}
   10440:   e28db004    add     fp, sp, #4
   10444:   e59f000c    ldr     r0, [pc, #12]   ; 10458 <main+0x1c>
   10448:   ebffffa5    bl      102e4 <puts@plt>
   1044c:   e3a03000    mov     r3, #0
   10450:   e1a00003    mov     r0, r3
   10454:   e8bd8800    pop     {fp, pc}
   10458:   000104cc    andeq   r0, r1, ip, asr #9
```