Лекция 15

Кодирование инструкций

- Каждая инструкция однозначно представляется в бинарном виде
- На x86/x64 инструкция кодируется последовательностью от 1 до 16 байт
- Требования к кодированию:
  - Однозначность декодирования
  - Компактность

Кодирование относительно IP

- Инструкция jmp:
  8048098:eb 17 jmp 80480b1
- Занимает 2 байта, адрес после нее: 804809A
- В инструкции кодируется смещение относительно EIP: 17
- 804809a + 17 = 80480b1
- На x86 так кодируются инструкции call, jmp, jCC
- На x64 существует специальный режим адресации: RIP-relative:
  mov L1(%rip), %rax

Абсолютное кодирование

- Загрузка адреса в памяти на регистр:

```
80480a2:  b9 b5 80 04 08    mov $0x80480b5,%ecx
80480b5:  48 65 6c 6c 00    .asciz “Hell”
```

- Загрузка значения глобальной переменной в регистр
- В закодированной инструкции записывается абсолютный адрес в памяти

Загрузка программы в память

- Программа – единое целое, взаимное расположение кода внутри секций и секций друг относительно друга не изменяется
  - Смещения в относительных переходах и работе с памятью настраиваются компоновщиком и при загрузке в память не изменяются
- При компоновке фиксируется адрес, по которому программа должна размещаться в памяти, абсолютные адреса в программе настраиваются относительно него
  - ELF Linux x86 по умолчанию: 0x804800
  - Можно изменить с помощью -Wl,-Ttext-segment=ADDR

Позиционная зависимость

- Если программа неработоспособна при загрузке по адресу, отличному от прописанного в исполняемом файле – программа **позиционно зависима**
- **Позиционно-независимый код** (PIC – position independent code) – сохраняет работоспособность при загрузке с любого адреса в памяти
- Полезен:
  - В динамических библиотеках
  - Динамическая кодогенерация