Когда защита стека небезопасна?

Довгалюк П. М., Макаров В. А.

Переполнение буфера в стеке

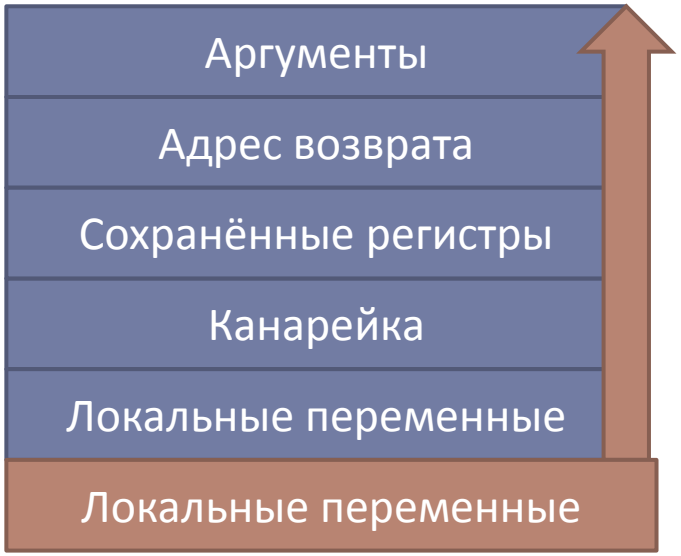
- Атака на
  - Адрес возврата
  - Локальные переменные
  - Сохраняемые регистры
  - Аргументы функции

Стековая канарейка

- Защищает адрес возврата от перезаписи
  - Также может защищать esp, ebp, и сохраняемые регистры

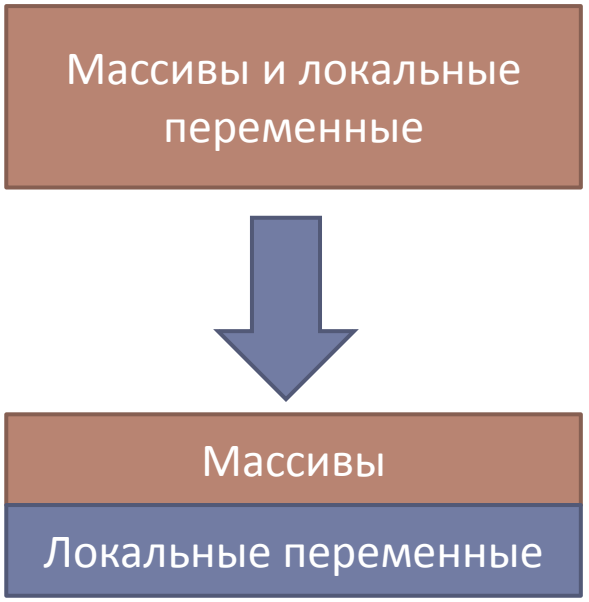
Когда стековая канарейка не срабатывает?

- Проверка значения делается только в конце функции
- Могут использоваться некорректные значения переменных и аргументов



Перемещение переменных

- Массивы располагаются выше в памяти
- Переполнение массивов не затронет обычные переменные

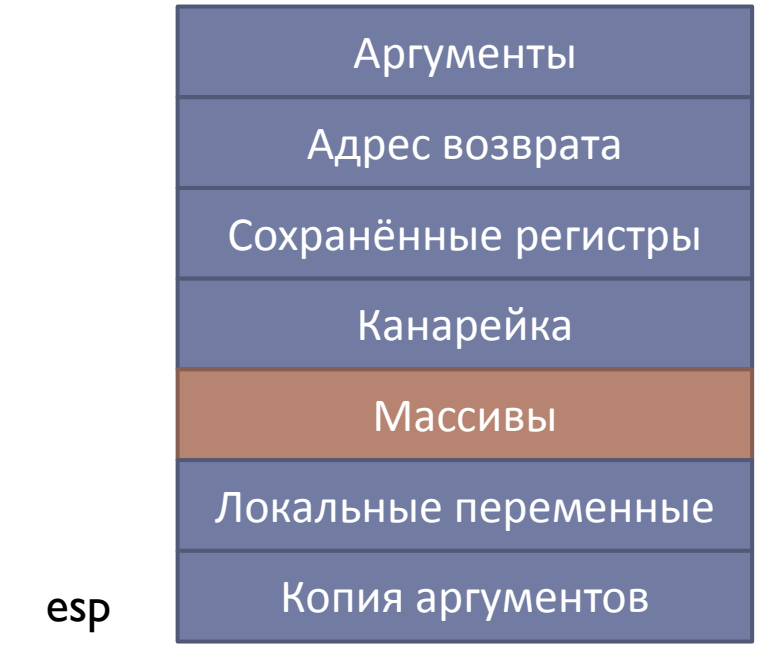


Копирование аргументов

- В случае переполнения буферов копия аргументов не портится
- Влияние на работу функции уменьшается

Цель исследования

- Есть ли особенности кодогенерации в современных компиляторах, которые не позволяют защищать программы с помощью анализа помеченных данных
- Побочный результат: ошибки кодогенерации при защите данных в стеке

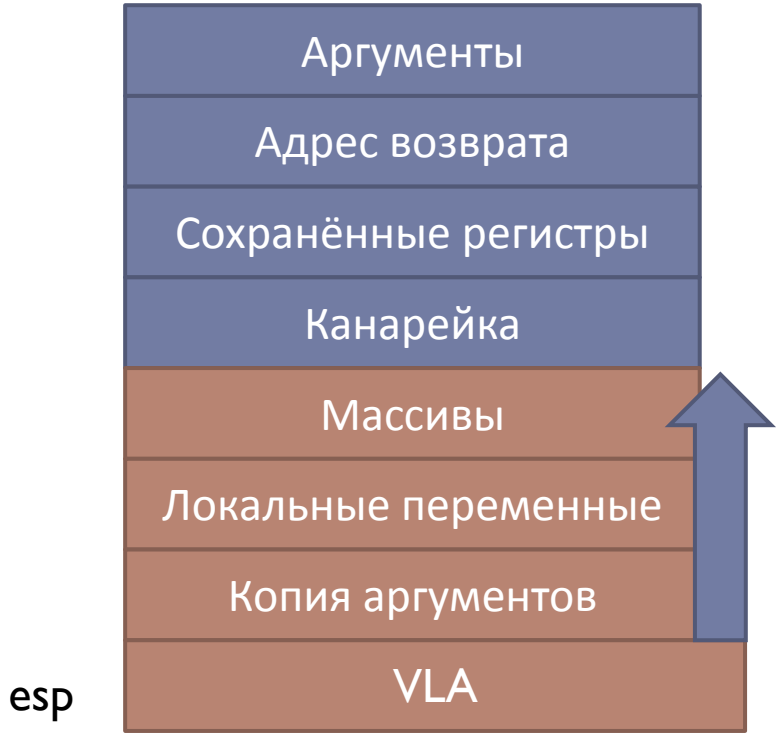


Тестирование компиляторов

- Набор исходных текстов + комбинации опций компиляции
- MSVC 2015
- gcc 5.2.0
- clang 3.7.1

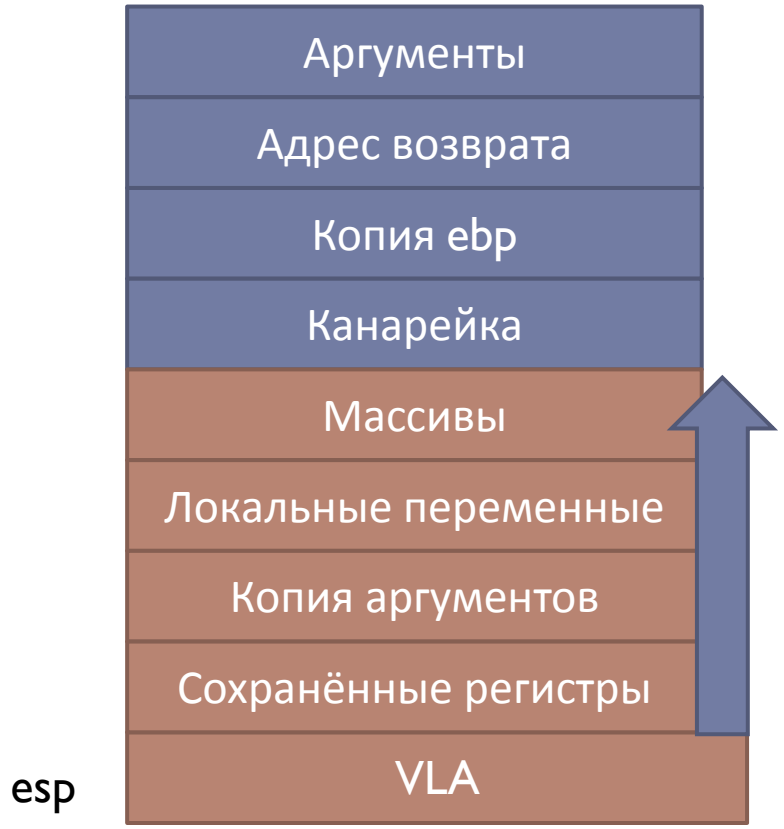
Защита стека и VLA

- VLA – массивы переменного размера
  - int n;
  - int a[n];
  - int *p = alloca(n);
- Аргументы и локальные переменные не защищены от переполнения VLA



MSVC

- MSVC не защищает сохранённые регистры с помощью канарейки
- Атакующий может управлять ими в случае переполнения VLA



Выравнивание локальных переменных

- Если указатель стека выравнивается, то сохраняется его копия
- Содержимое регистра-копии ничем не защищено
- Он может быть испорчен при вызове незащищённой функции

MSVC: Выравнивание локальных переменных

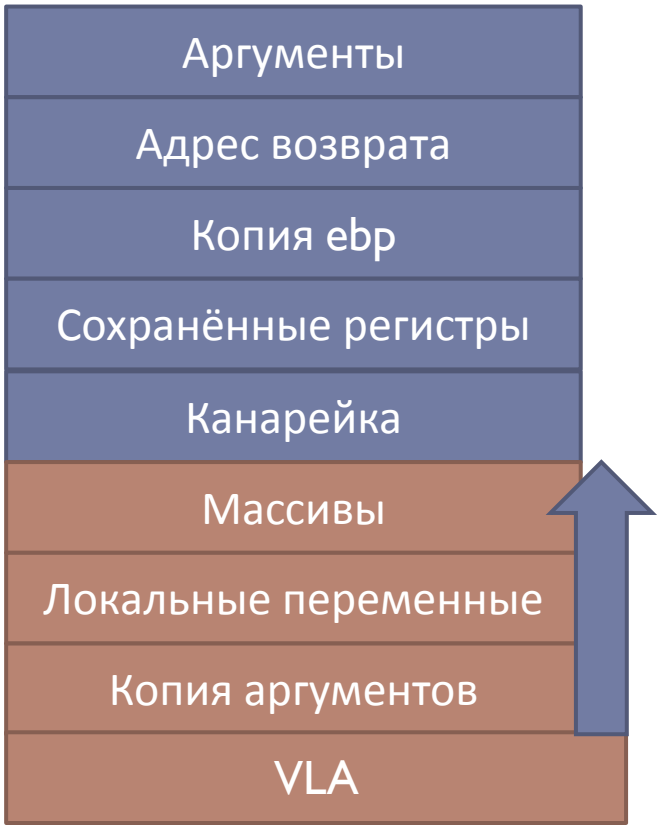
```
push ebx
mov ebx, esp
sub esp, 8
and esp, -32
add esp, 4
push ebp
...
mov ecx, [ebp-4]
xor ecx, ebp
call @__security_check_cookie@4
mov esp, ebp
pop ebp
mov esp, ebx
pop ebx
ret 0
```

clang: Выравнивание локальных переменных

```
push ebp
mov ebp, esp
push ebx
push edi
push esi
and esp, 0xfffffff0
sub esp, 0x60
mov esi, esp
...
mov eax, ___stack_chk_guard
cmp eax, [esi+0x48]
jne L
lea esp, [ebp-0xc]
...
```

GCC

- Массивы переменного размера могут затереть только другие переменные



Методы защиты, не вошедшие в компиляторы

- Проверка переменной-канарейки перед вызовами функций и после них
- Своя канарейка для каждого буфера
- Вероятностное расположение канарейки в стеке

Заключение

- Стековая канарейка не всегда справляется с особенностями компиляторов
- Исходные тексты и скрипты для компиляции есть на github
- Дальнейшая работа – применим ли динамический анализ потоков данных для защиты от переполнения буфера