Разработка безопасного компилятора на основе Clang

Павел Дунаев, Артем Синкевич, Артемий Гранат, Инна Батраева, Сергей Миронов, Никита Шугалей

ИСП РАН, СГУ

2024

Безопасный компилятор: основные функции

- Безопасные оптимизации (не вносят дополнительные уязвимости);
- Оповещение о наличии неопределённых конструкций;
- Динамический контроль неопределённого поведения;
- Управление распределением памяти;
- Прочие механизмы защищённости.

При этом:

- Правки, требуемые для успешной компиляции минимально возможны;
- Не предоставляется возможность выборочного отключения большинства возможностей.
- Возможности разделены на 3 класса безопасности.

Безопасная компиляция с помощью Clang: 3 класс

Отключение преобразований, связанных с...

Требование		Опция
Целочисленным переполнением		`-fwrapv`
Тем фактом, что значения указателей разных типов могут совпадать		`-fno-strict-aliasing`
Разыменованием `nullptr`		`-fno-delete-null-pointer-checks`
Правым аргументом побитовых сдвигов	+	`-fkeep-oversized-shifts`
Делением на ноль и взятием нулевого остатка	+	`-fkeep-div-by-zero`

New feature: -fkeep-oversized-shifts

```
int testShl(void) {
  int Base = input();
  if (Base >= 1 && Base <= 6) {
    int Result = 1 << Base;
    if (Result == 0) return 5;
    return Result;
  }
  return -1;
}
```

```
int testShl(void) {
  int Base = input();
  int Result = 1 << Base;
  if (Result == 0) return 5;
  return Result;
}
```

Unsafe: не может вернуть 5.
Safe: может вернуть 5 на некоторых архитектурах.

New feature: -fkeep-div-by-zero

```
int foo(int x) {
    return x / 0 ? 1 : 2;
}
```

Unsafe: только `retq`.
Safe: полный код со сравнением.

Безопасная компиляция с помощью Clang: 3 класс

Требование		Опция
Защита от переполнения буфера при вызове некоторых функций	+	`-D_FORTIFY_SOURCE=2` (3 на `-Safe[1..2]`); fortify headers
Контроль за целостностью стека		`-fstack-protector-strong`, `-fstack-clash-protection`
Генерация позиционно-независимого кода		`-fpic`/`-fPIC`/`-fpie`/`-fPIE`, `-pie`
Запрет на inline некоторых функций	+	`-fno-builtin-*` + fortify headers

Безопасная компиляция с помощью Clang: 3 класс

Предупреждение о...

Требование		Опция
Чтении или записи по некорректному индексу		`-Warray-bounds`, `-Warray-bounds-pointer-arithmetic`
Целочисленном делении на ноль или взятии нулевого остатка		`-Wdivision-by-zero`
Побитовом сдвиге с некорректным правым операндом		`-Wshift-count-negative`, `-Wshift-count-overflow`
Перезаписи локальной переменной при вызове `longjmp`	+	Опция `-fforce-volatile-before-setjmp` вместо предупреждения

Безопасная компиляция с помощью Clang: 2 класс

Требование		Опция
Сохранение побочных эффектов записи в память	+	`-fpreserve-memory-writes`
Автоматическая инициализация переменных нулями		`-ftrivial-auto-var-init=zero`
Запрет оптимизаций побитовых сдвигов с учётом некорректного правого аргумента	+	`-fkeep-oversized-shifts`
Использование операций с векторными машинными регистрами, не требующих выравнивания данных	+	`-fassume-unaligned`
Запрет оптимизаций адресной арифметики по информации о размерах объектов		**WIP**
Остановка компиляции с ошибкой для предупреждений 3 класса		`-Werror=*`
Запрет использования функции `gets`	+	fortify headers

Безопасная компиляция с помощью Clang: 1 класс

Требование		Опция
Динамический контроль неопределённых конструкций	↑	UBSan: `-fsanitize=*`, `-fsanitize-trap=*`
Уникальное распределение статической памяти на этапе компиляции	+	`-frandom-func-reorder`, `-frandom-func-and-globals-reorder`
Рандомизация автоматической памяти	+	`-floc-var-per` (+ `-fadd-loc-var`)
Динамическое распределение памяти		поддерживает по умолчанию при подходящих компоновщике и загрузчике

Новые и переработанные санитайзеры

Санитайзер		Комментарий
`float-to-float-cast-overflow`	+	Проверка переполнения при преобразовании float to float
`null-call`	+	Проверка на `nullptr` при непрямом вызове функции
`function`	↑	+ поддержка C и новых архитектур
`return-c`	+	Поддержка `return` для C

Тесты производительности

Тест	Baseline	-Safe3	-Safe3 замедл.	-Safe2	-Safe2 замедл.	-Safe1	-Safe1 замедл.
GNU Go	3.93 с	4.03 с	2.54%	4.26 с	8.12%	6.66 с	69.04%
LAME	5.21 с	5.27 с	1.15%	4.90 с	-5.82%	13.19 с	153.40%
fannkuch	2.24 с	2.10 с	-6.25%	2.08 с	-7.14%	2.72 с	21.43%
x264	1.69 с	1.81 с	7.42%	1.79 с	5.20%	6.32 с	**274.74%**
zlib	1.54 с	1.63 с	5.88%	1.62 с	5.87%	2.40 с	55.60%

AMD Ryzen™ 5 4600H (x86-64), Manjaro Linux 23.1.4.

Сборка пакетов Alpine Linux

Класс	Успешно собрано	Ошибки сборки
Baseline	3587	910
`-Safe3`	3581	6
`-Safe2`	3500	81
`-Safe1`	3456	44

Q & A