Идентификация реквизитов сборки через отслеживание системных вызовов

Артемий Гранат, Павел Дунаев, Артем Синкевич,
Инна Батраева, Дмитрий Петров

Институт системного программирования РАН им. В.П. Иванникова
Саратовский государственный университет

21 июня 2024 г.

Актуальность

- 1 апреля 2024 года был введён государственный стандарт «ГОСТ Р 71206-2024 Защита информации. Разработка безопасного программного обеспечения. Безопасный компилятор языков C/C++. Общие требования»;
- Одним из пунктов стандарта является функция журналирования процесса трансляции, с сохранением такой информации, как параметры компиляции и хэш-суммы входных и выходных файлов в формате JSON.

Обзор существующих инструментов

Инструмент	Командная строка
Bear	`bear -- gcc main.c`
compiledb	`compiledb gcc main.c`
Clang	`clang main.c -MJ compile_commands.json`
SAFEC	`<safec> --dump-sbom <dir> main.c`
CMake	`cmake (CMAKE_EXPORT_COMPILE_COMMANDS ON)`
qmake	`qmake (QMAKE_GENERATE_COMPILE_COMMANDS = yes)`

Альтернативные подходы

- FUSE
- inotify
- ptrace

FUSE

- Позволяет создавать собственные файловые системы в пользовательском пространстве, что в свою очередь позволяет реализовывать собственные механизмы отслеживания и журналирования операций над файлами, связанных с процессом сборки
- Не выдает информацию о командах сборки
- Может работать медленнее, чем файловая система, реализованная на уровне ядра ОС

inotify

- Позволяет отслеживать действия, связанные с файлами и директориями, с которыми происходит взаимодействие в процессе сборки
- Как и в случае с FUSE, не выдает информацию о вызываемых командах сборки
- `max_user_watches`

ptrace

- Позволяет получить информацию об открытых файлах, командах сборки, а также о созданных процессах
- Необходимо учитывать зависимости от процессорной архитектуры при разработке инструмента
- Один процесс может быть отслеживаем только одним трассировщиком в момент времени

Наиболее релевантным подходом для разработки инструмента, решающего задачу сбора информации о процессе сборки является ptrace, так как он обеспечивает полный контроль над исполнением процесса сборки и собирает подробную информацию о его действиях.

Инструмент sbom-trace

Инструмент sbom-trace состоит из двух частей: трассировщика системных вызовов и постпроцессора.

- Компонент sbom-tracer является ptrace-монитором для всего дерева процессов сборки и перехватывает системные вызовы семейства open (и только их) через seccomp-фильтр
- sbom-postprocessor обрабатывает и дополняет данные, полученные из трассировщика

Отслеживаемые системные вызовы

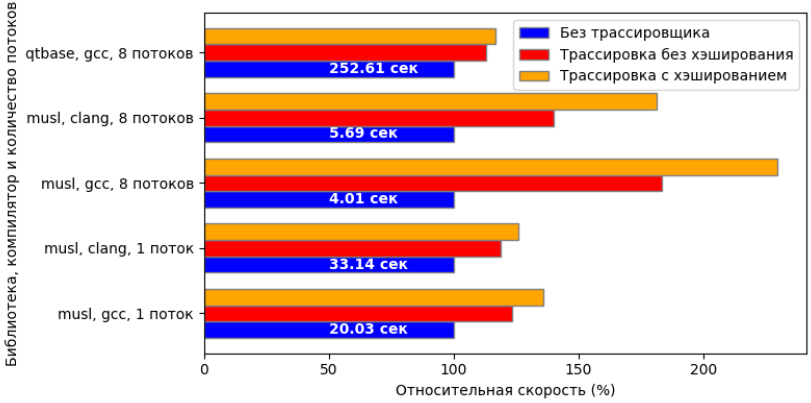
Системный вызов	Необходимая информация
open, openat, openat2	Путь к файлу и режим, в котором он был открыт
execve	Список аргументов команды
fork, vfork, clone	Идентификатор нового процесса

Итоговая база данных компиляции

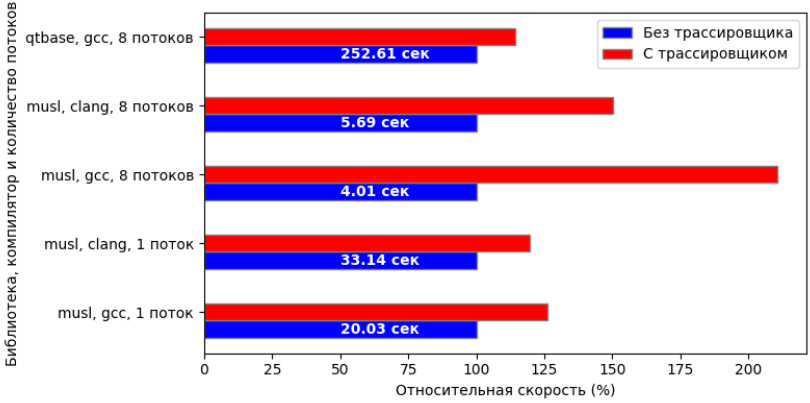
- directory
- input
- command
- component_commands
  - command
  - dependencies
  - output
- utilities

```
$ sbom-tracer --postprocessor sbom-postprocessor.py --filter-subtree
    /root gcc main.c
{
    "directory": "/root",
    "input": { "/root/main.c": "..." },
    "commands": [ "gcc", "main.c" ],
    "component_commands": [
        {
            "command": [ "/usr/lib/gcc/x86_64-linux-gnu/11/cc1", "...", ],
            "dependencies": { "/etc/ld.so.cache": "...", },
            "output": { "/tmp/ccUEQNfb.s": "..." }
        },
        ...
    ],
    "utilities": [
        { "path": "/usr/bin/x86_64-linux-gnu-gcc-11", "hash": "..." }, ...
    ]
}
```

Измерение производительности однопоточного трассировщика



Измерение производительности многопоточного трассировщика



Q & A