докладчик:
Благодаренко А. В.

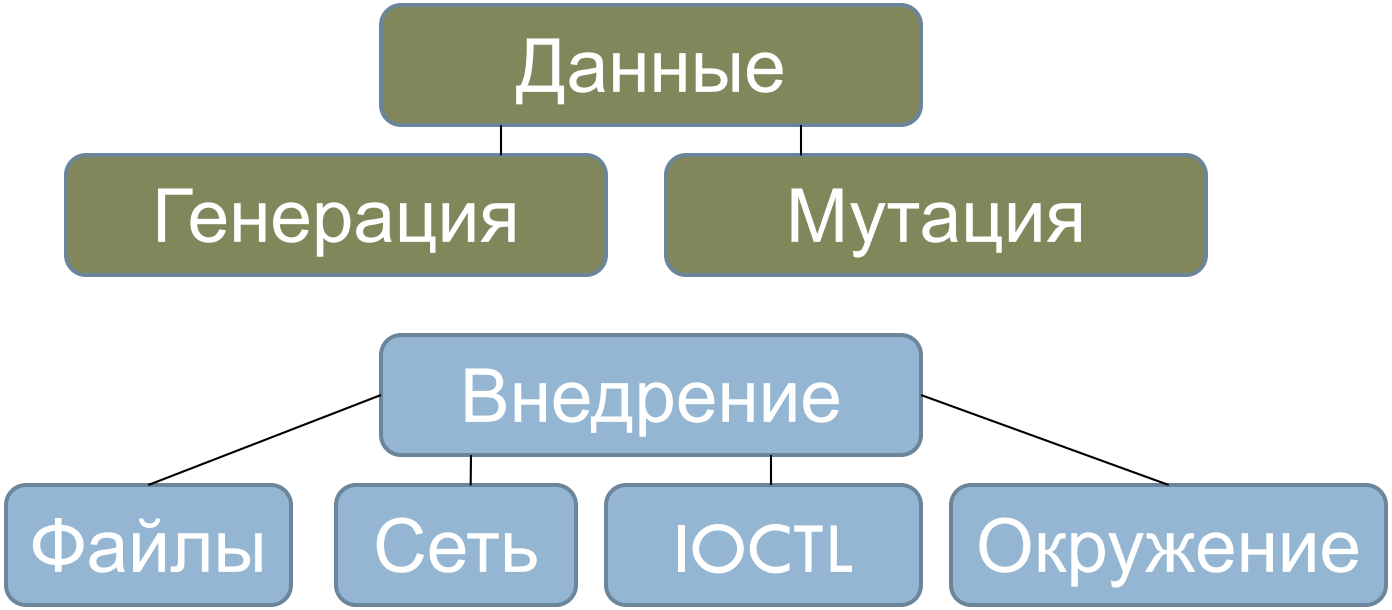
руководитель:
д.т.н., проф. Макаревич О. Б.

ВЫБОР ТОЧКИ ВНЕДРЕНИЯ ДЛЯ ФАЗЗИНГА В ПАМЯТИ

Технологический институт «Южного федерального университета» в г. Таганроге

Фаззинг

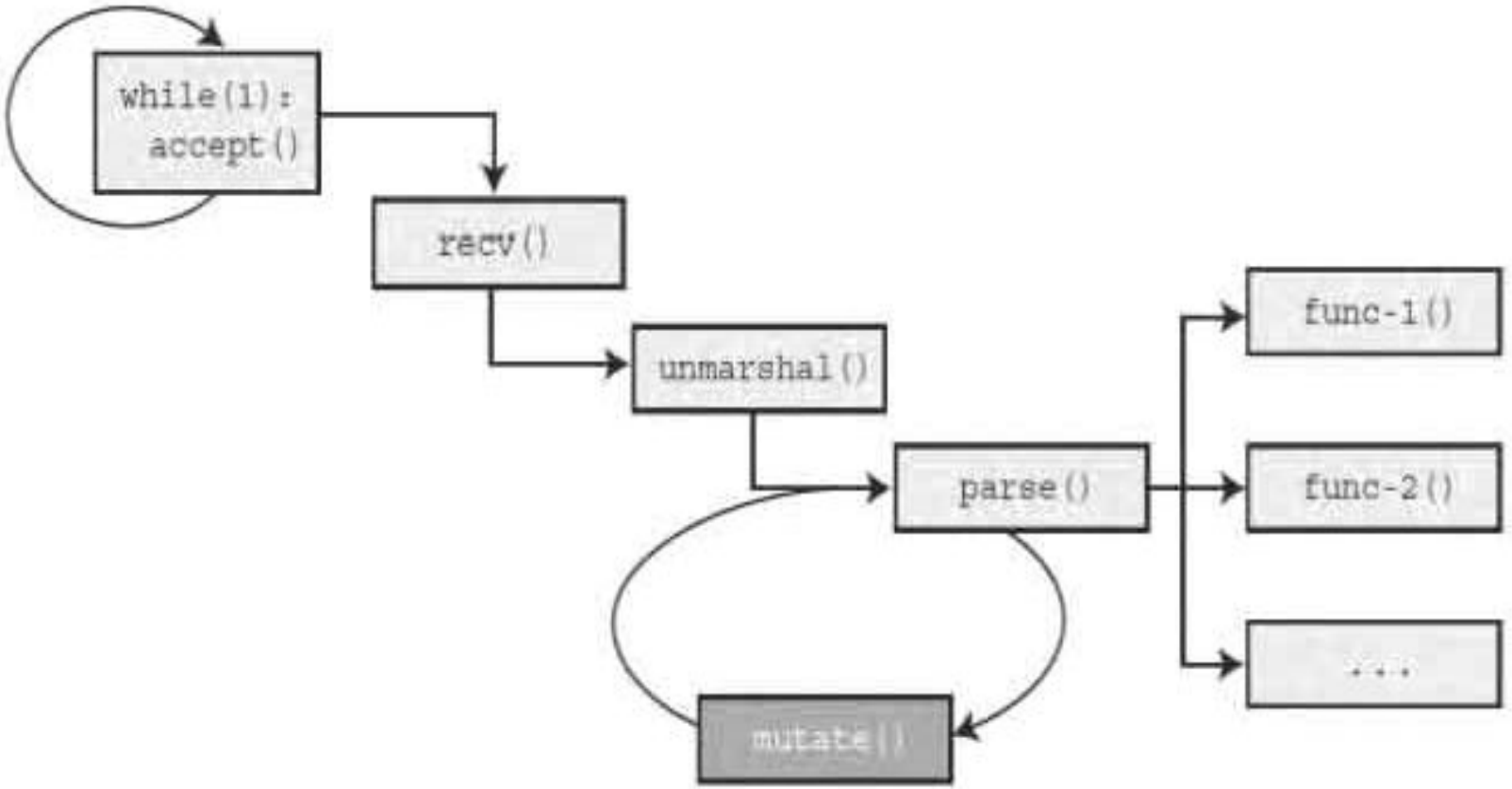
- Фаззинг – тестирование методом черного ящика, основанное на передаче большого набора входных данных исследуемому ПО



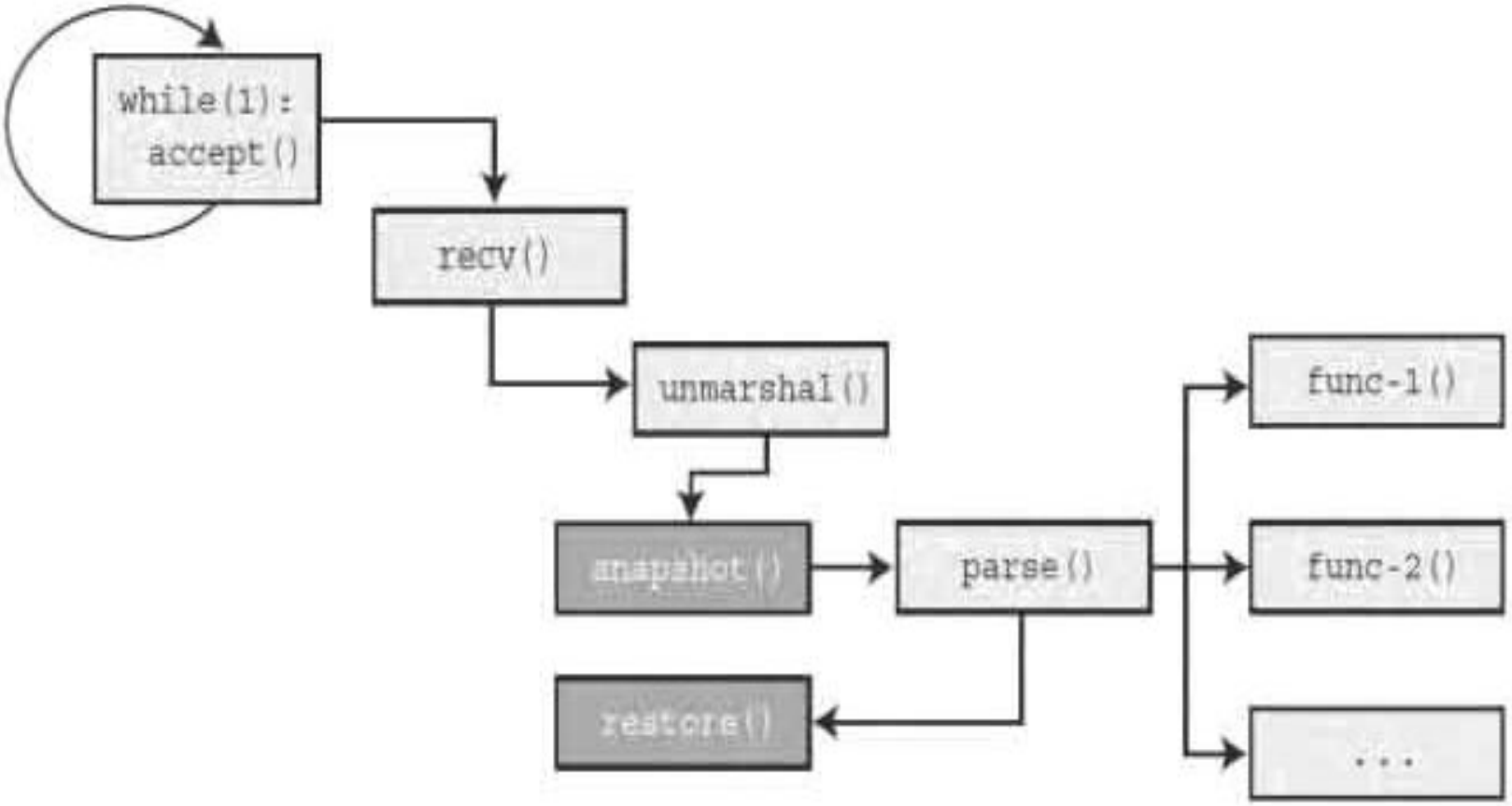
Фаззинг в памяти

- Фаззинг в памяти - вид фаззинга при котором данные передаются через ***внутренние структуры программы***
- Позволяет миновать интерфейсные функции программы и их ограничения (скорость, объемы данных).
- Позволяет сфокусироваться на данных, обрабатываемых исследуемой частью кода. Не требуется подготовка данных в формате, который требуется интерфейсными функциями.

Цикл мутации



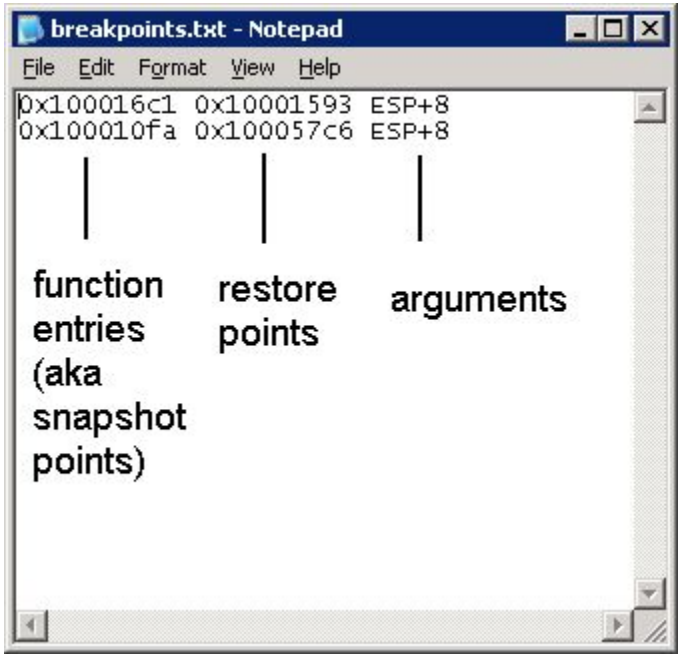
Восстановление состояния



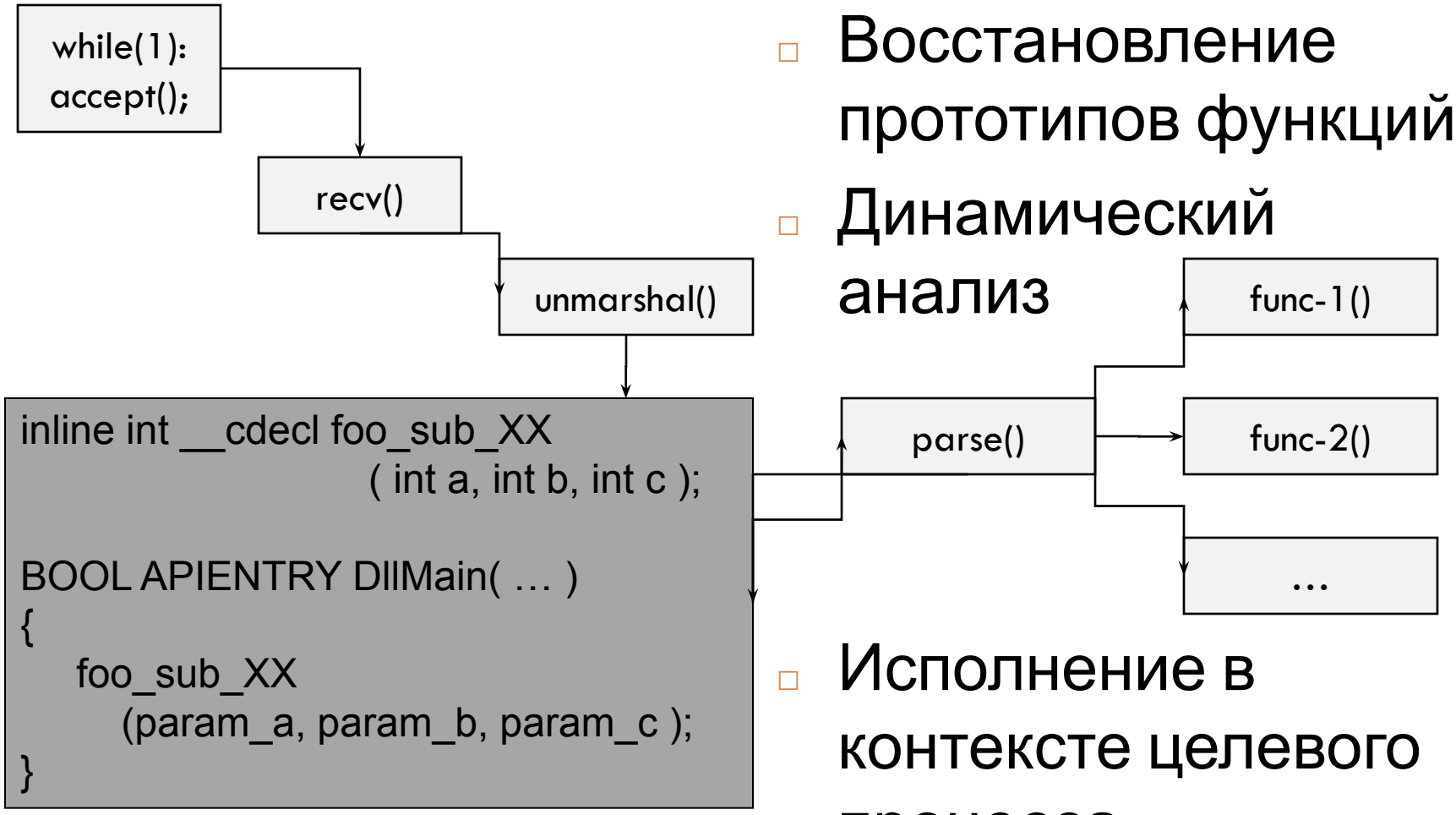
Corelan in-memory fuzzer

Необходимы данные:

- Целевой процесс/модуль
- Адрес точки входа в функцию (получение слепка)
- Адрес точки выхода из функции (восстановление слепка)
- Аргументы функции

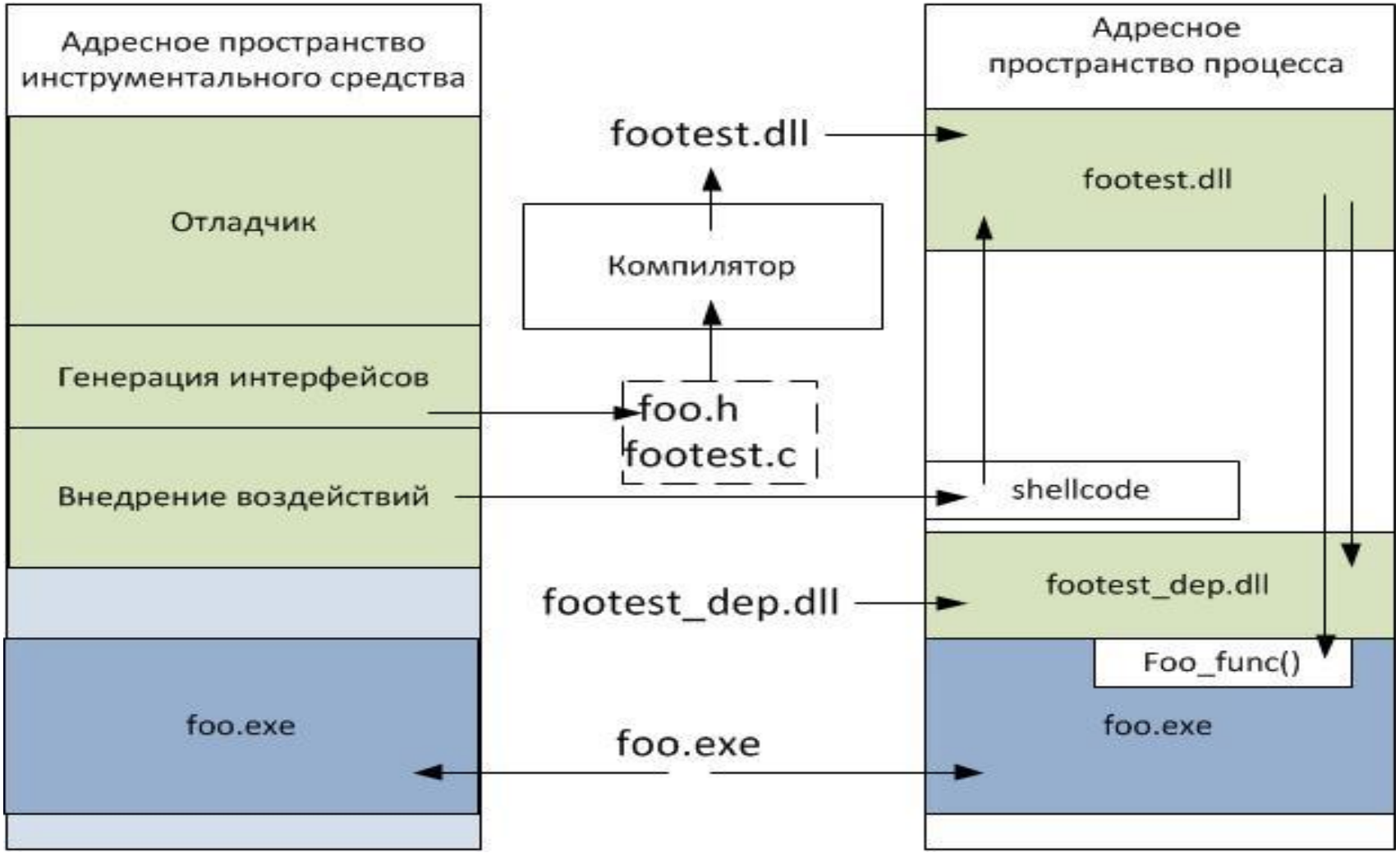


Фаззинг и модульные тесты

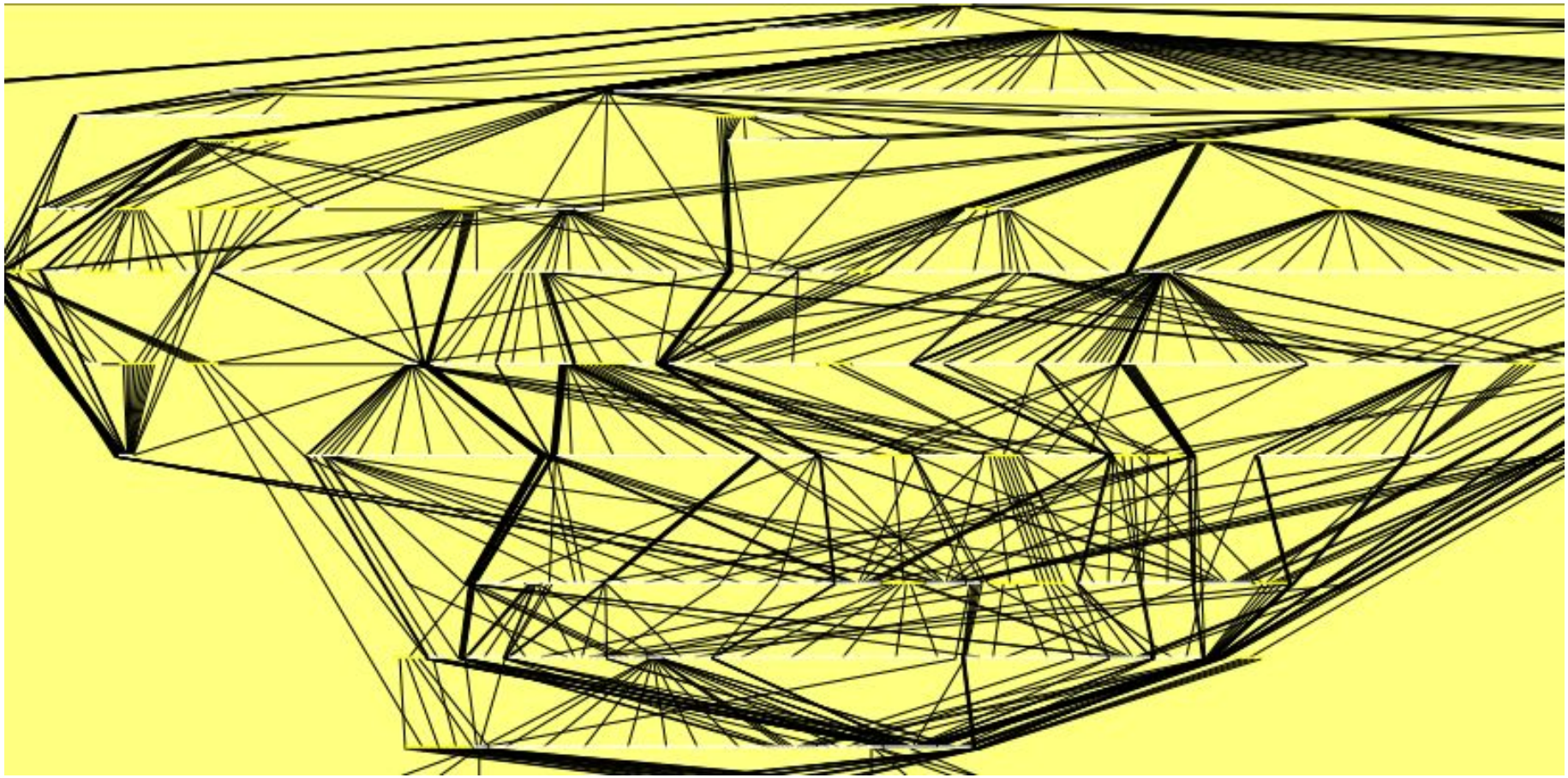


- Восстановление прототипов функций
- Динамический анализ
- Исполнение в контексте целевого процесса

Система Dataflow



Откуда начинать?



Рейтинг функций

Простой ориентированный граф G(V,E)

Матрица смежности $E = (e_{i,j})_{n \times n}$, где

$$e_{i,j} = 1 \Leftrightarrow (i,j) \in E$$

Матрица E дает информацию обо всех путях длины 1 в графе G(V,E). Композиция отношения E самой с собой

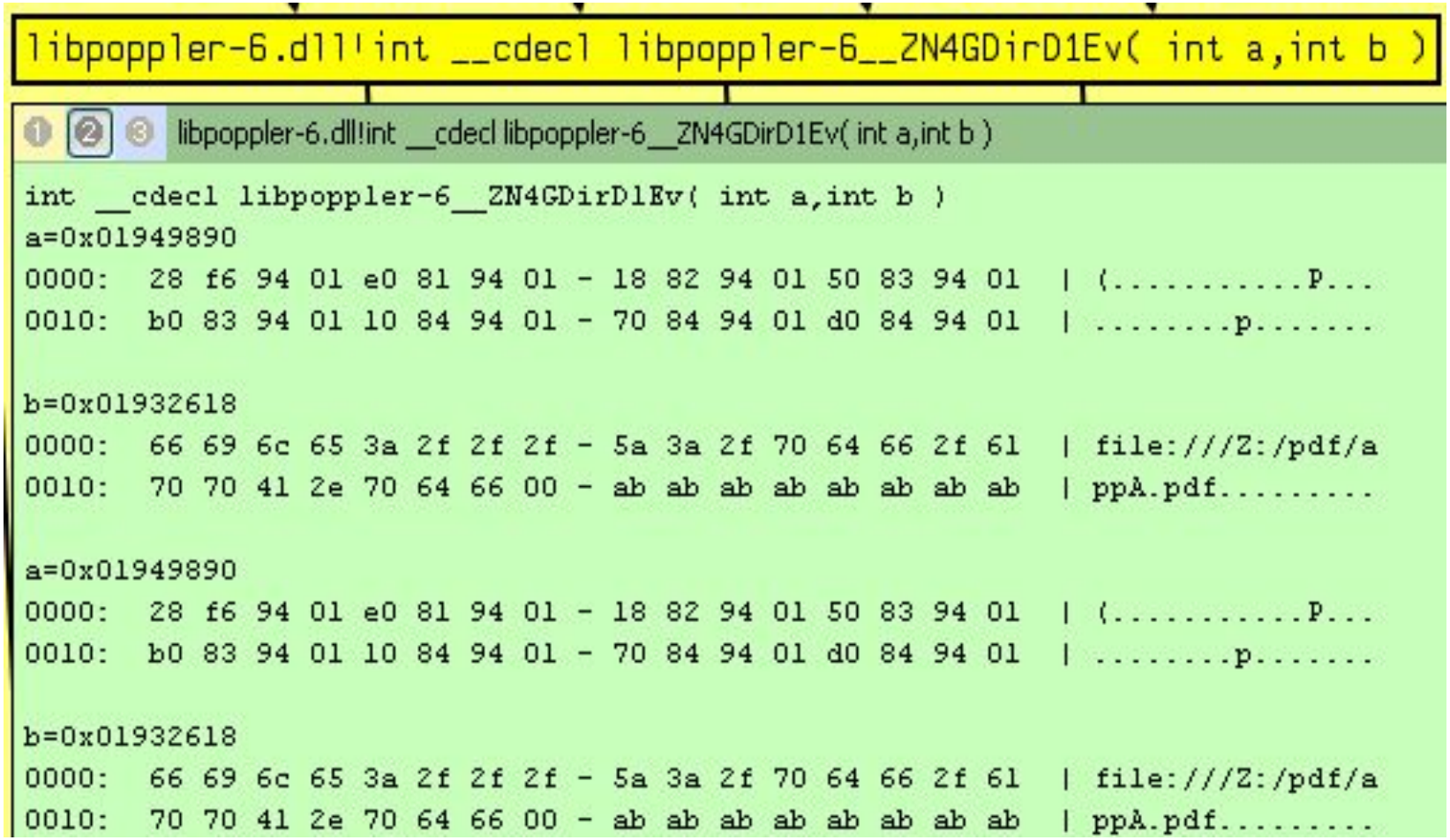
$$E \circ E = \{(a,c) : \exists b \in V : (a,b), (b,c) \in E\}$$

$$E^2 = e^2_{i,j}) = (\sum_{k=0}^{n} e_{ik} e_{kj}) = ((e_{i0} \wedge e_{0j}) \vee (e_{i1} \wedge e_{1j}) \vee .. \vee (e_{in} \wedge e_{nj}))$$

Матрица достижимости

$$E^* = E^1 \vee E^2 \vee .. \vee E^n = \left(e^*_{ij}\right)_{n \times n} = (e_{ij} \vee e^2_{ij} \vee .. \vee e^n_{ij})$$

Параметры функций

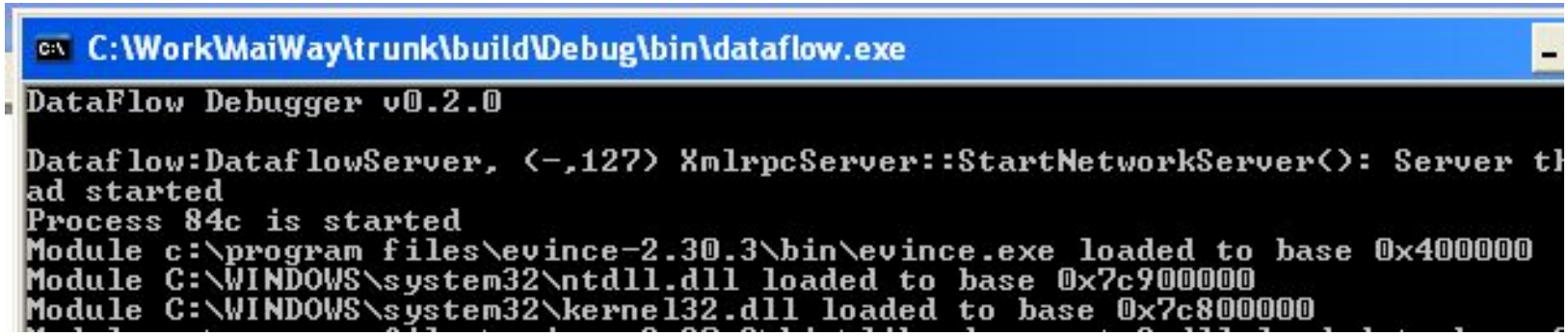


Пример

Фаззер файлов .pdf для программы Evince

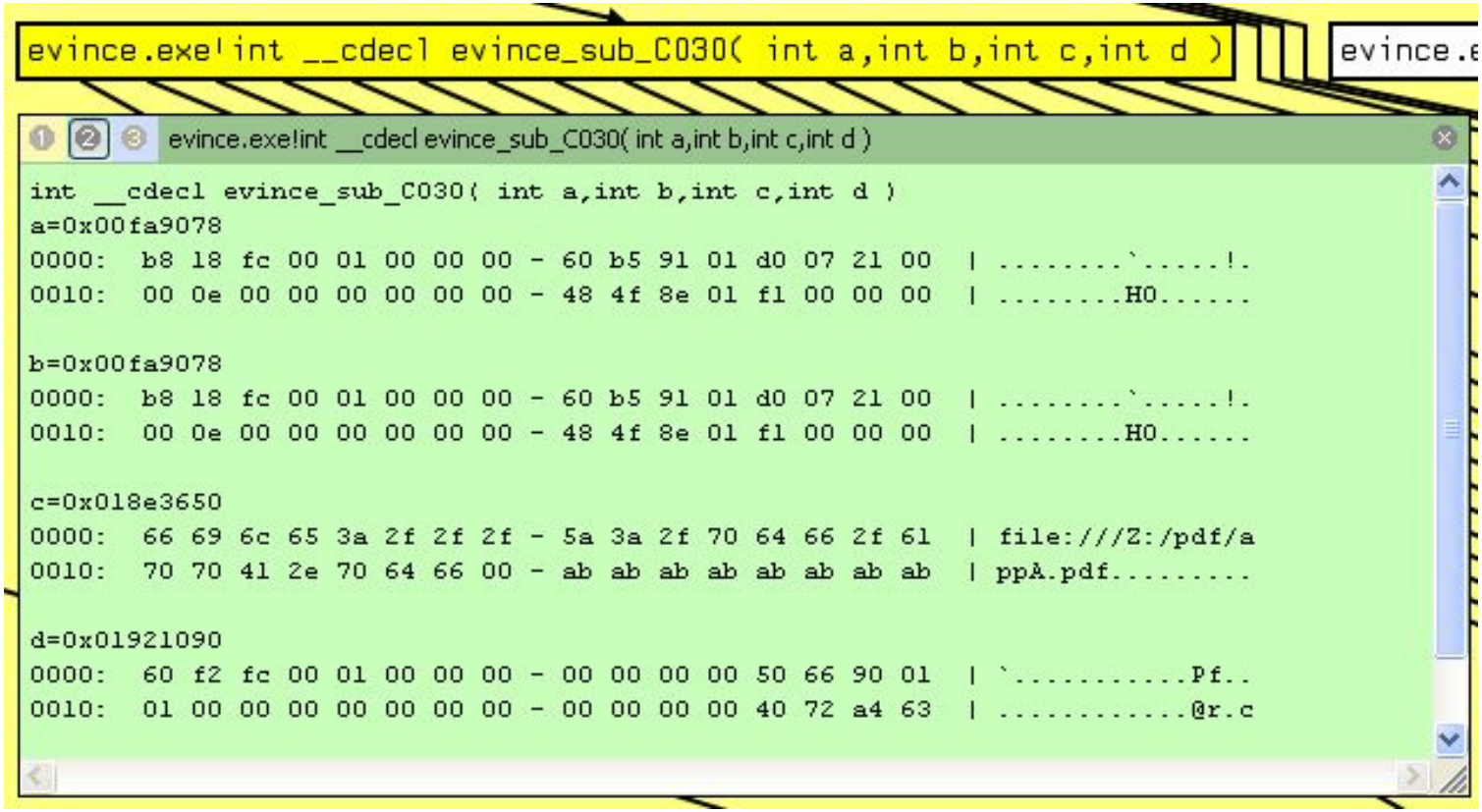
Тестовый запуск приложения

- Начать исследование
- Тестовый запуск 1
- Тестовый запуск 2
- Закончить исследование
- Получить статистику исполнения



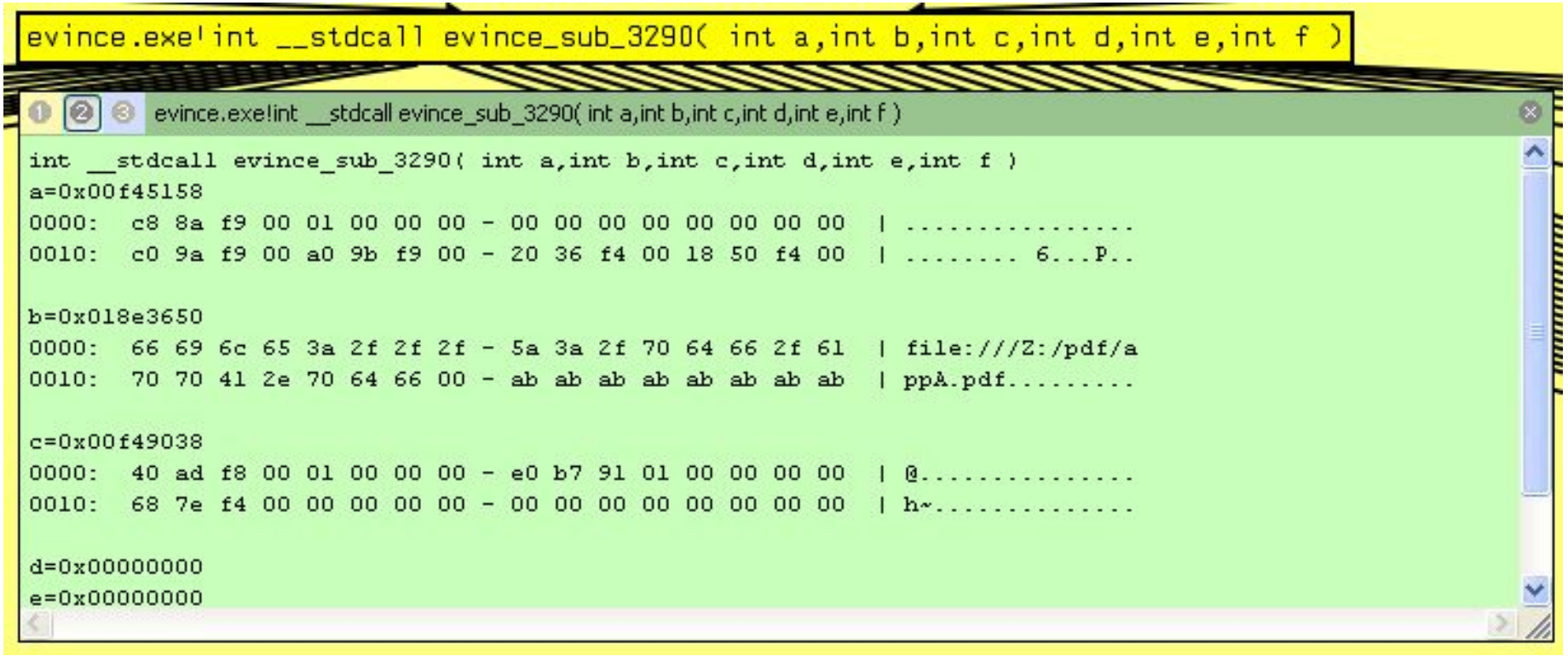
Функции, принимающие данные

Evince: 0xC030

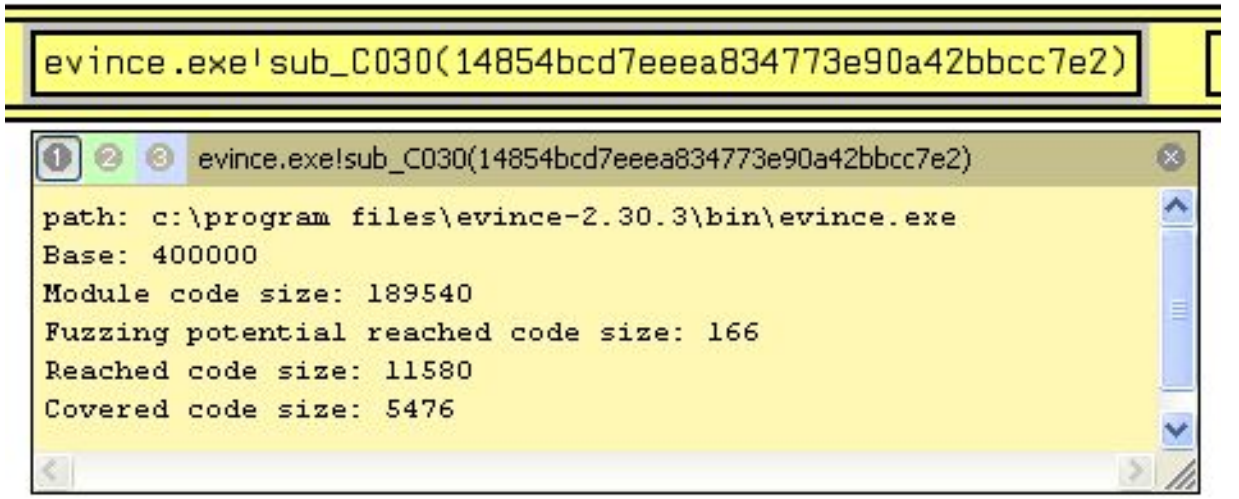


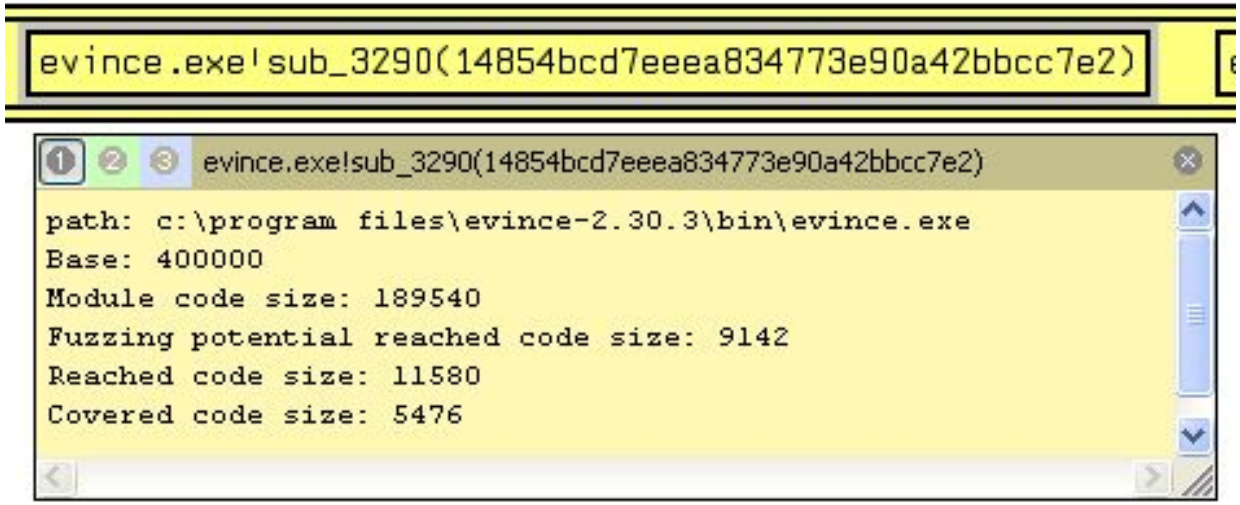
Функции, принимающие данные

Evince: 0x3290

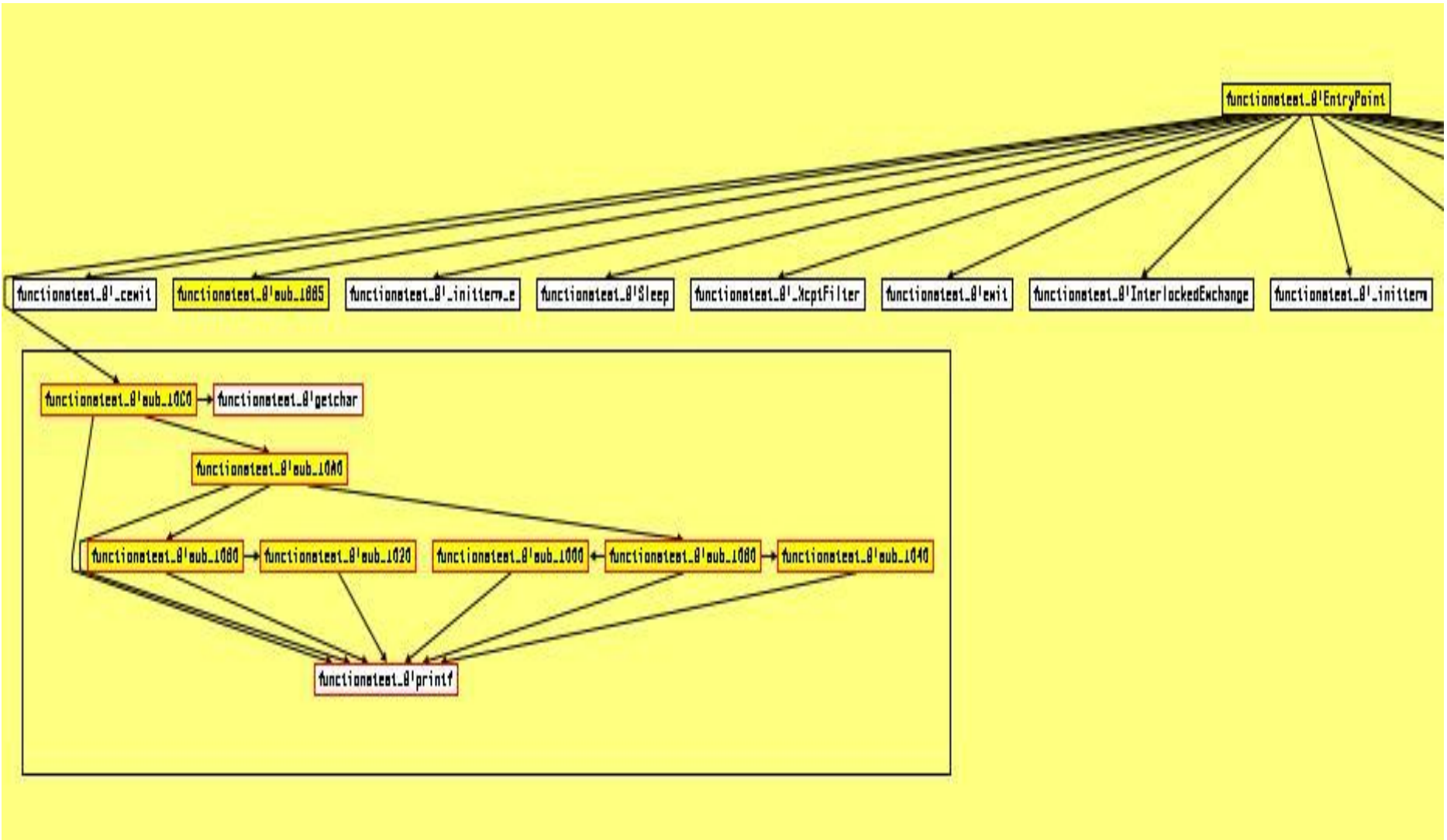


Рейтинг функций





Оценка потенциального охвата



Подготовка теста. Возможности

- Вызов внутренних функций с заданными параметрами
- Динамическая оценка покрытия
- Последовательное внедрение ошибок
- Любые другие возможности, применимые к динамическим библиотекам

Подготовка теста

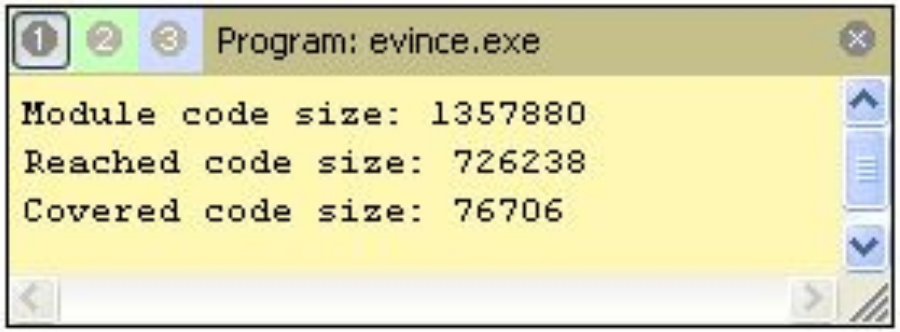

```
#include "evince.h"

CHECK_WITH_FAULT_INJECT( evince_sub_3290( 0x00f45158, (
    int )fileName, 0x00f49038, 0, 0, 0 ) );


if( SendCommand( COMMAND_TRACK_STAT, ffd.cFileName ) )
{
    LogErr( "Can't send command\n" );
}
```

Исполнение теста

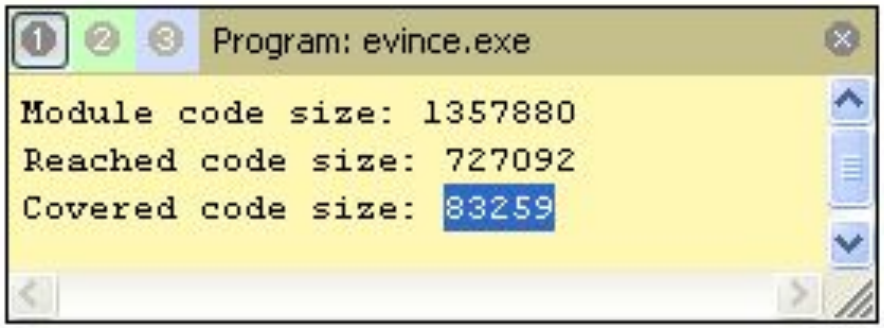
- Остановка на определенном этапе работы исследуемого ПО
- Загрузка динамической библиотеки в адресное пространство исследуемого ПО
- Исполнение
- Взаимодействие с тестирующим ПО:
- Внедрение ошибок
- Динамическая оценка покрытия

Оценка результатов


```
Program: evince.exe
```




```
Program: evince.exe
```



Вопросы?