

докладчик:

Благодаренко А. В.

руководитель:

д.т.н., проф. Макаревич О.
Б.

ВЫБОР ТОЧКИ ВНЕДРЕНИЯ ДЛЯ ФАЗЗИНГА В ПАМЯТИ

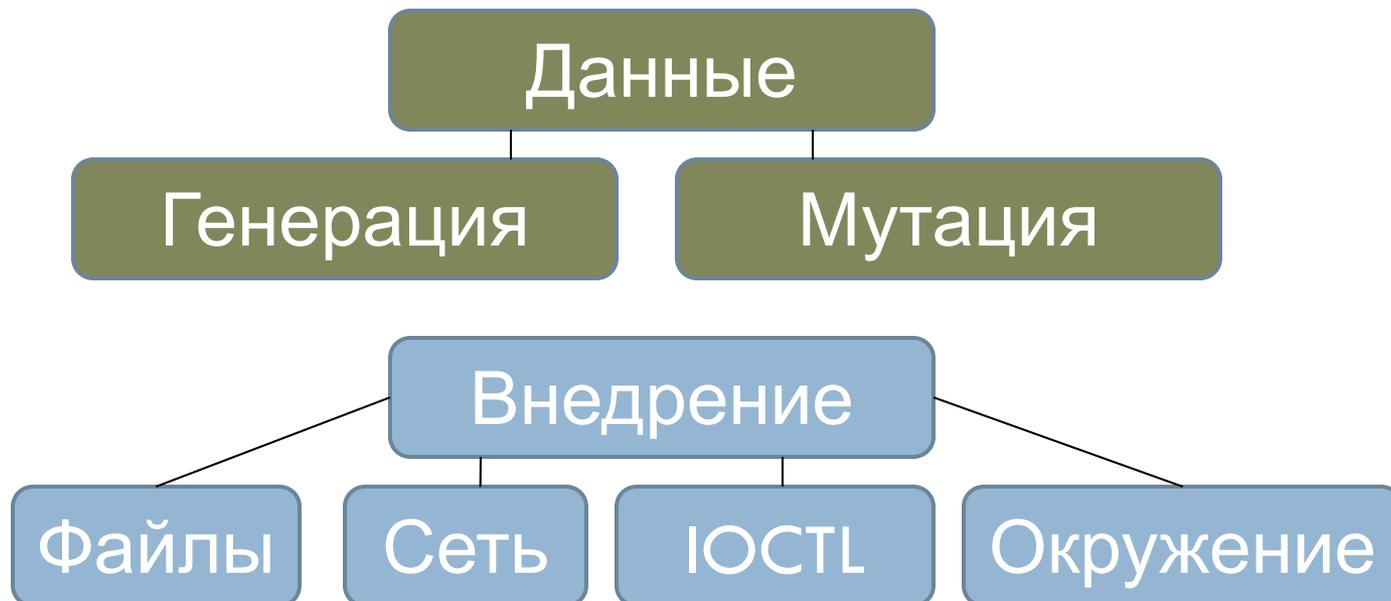


Технологический институт «Южного
федерального университета» в г. Таганроге

Фаззинг

2

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



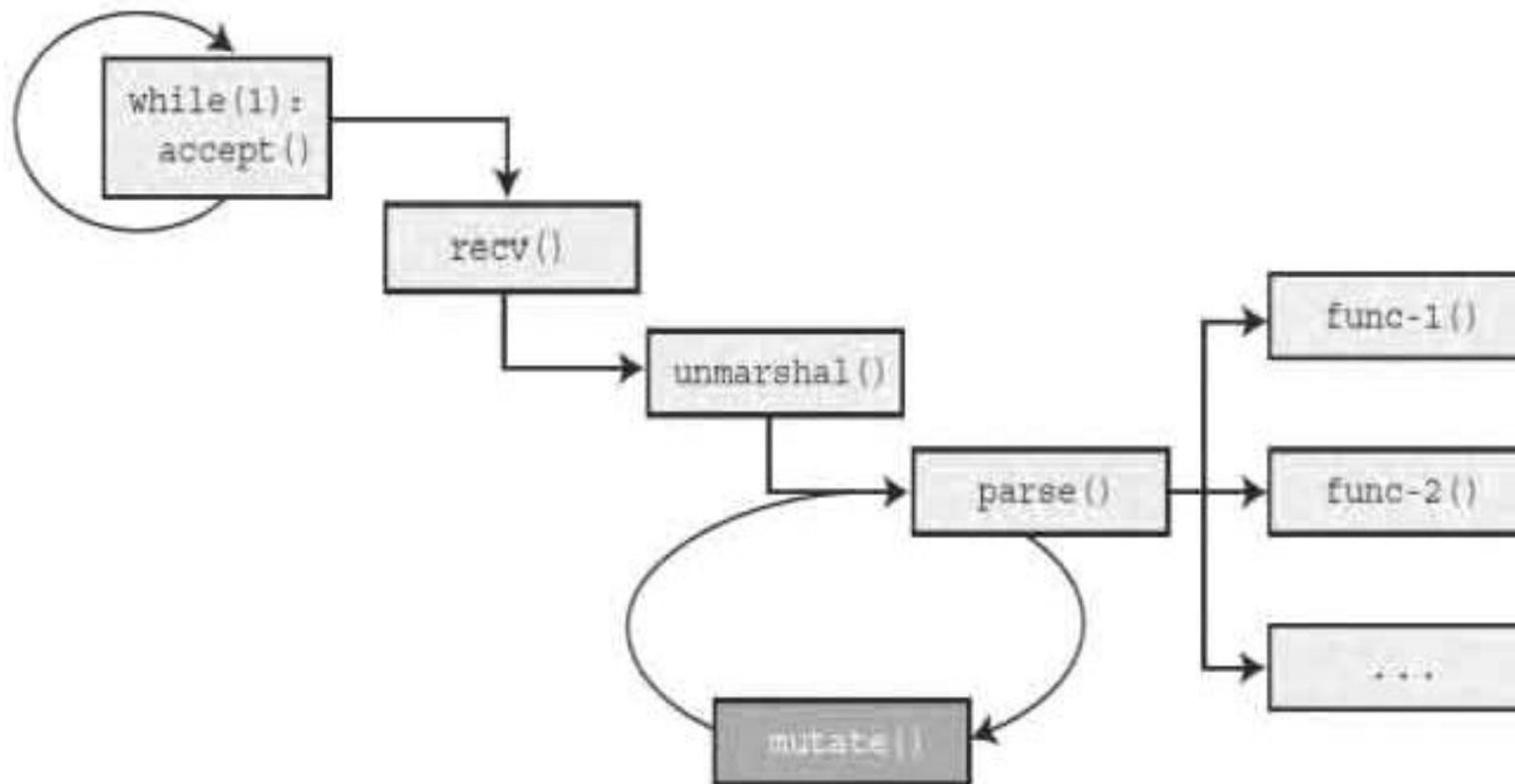
Фаззинг в памяти

3

- 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.

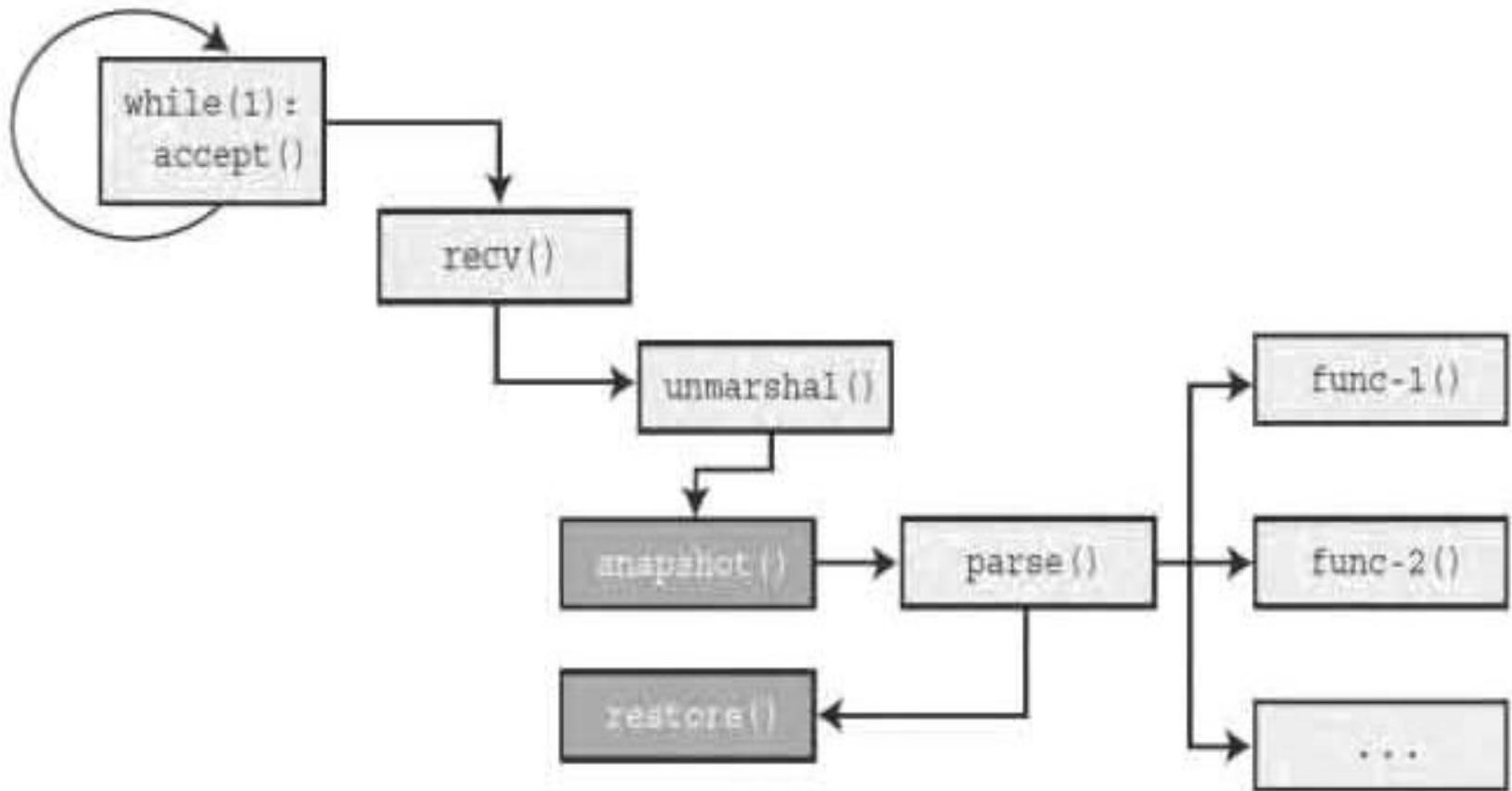
Цикл мутации

4



Восстановление состояния

5



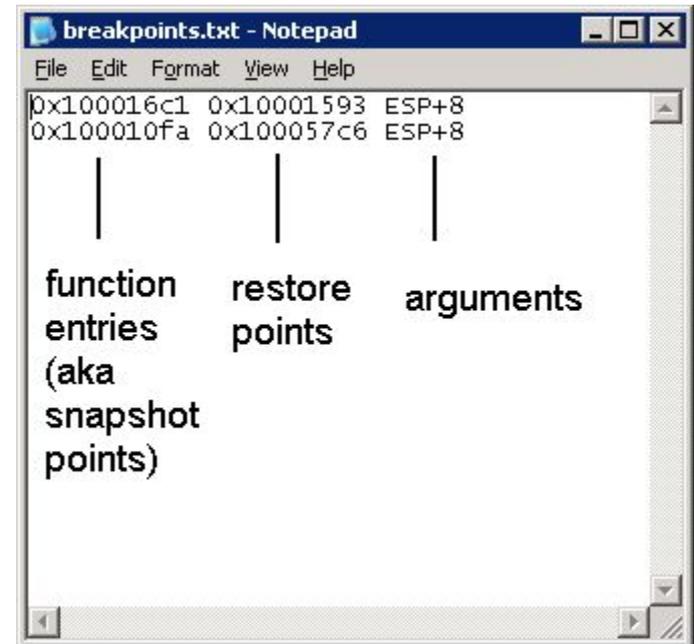
Corelan in-memory fuzzer

6

Необходимы данные:

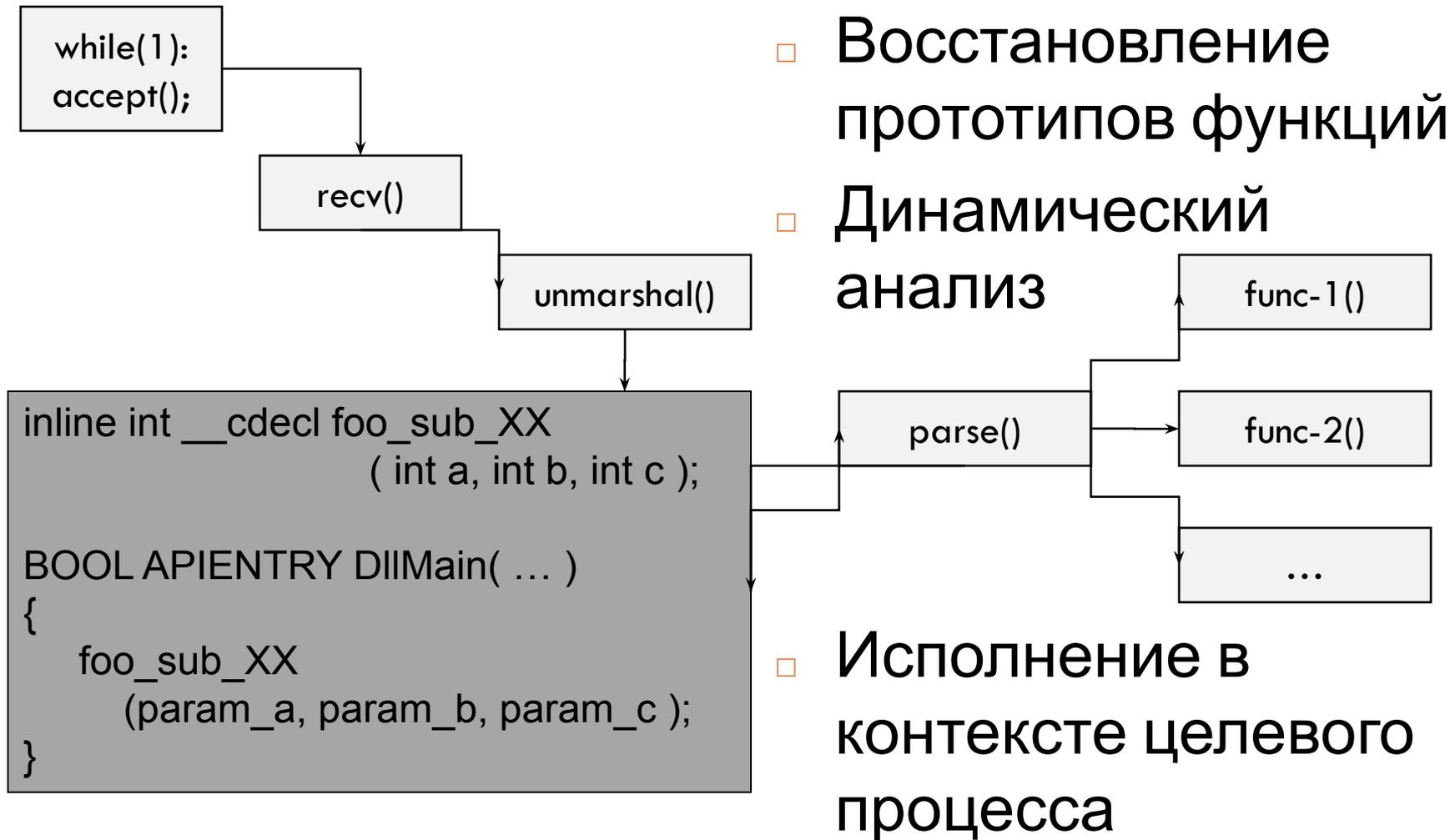
- Целевой процесс/модуль
- Адрес точки входа в функцию (получение слепка)
- Адрес точки выхода из функции (восстановление слепка)

- Аргументы функции



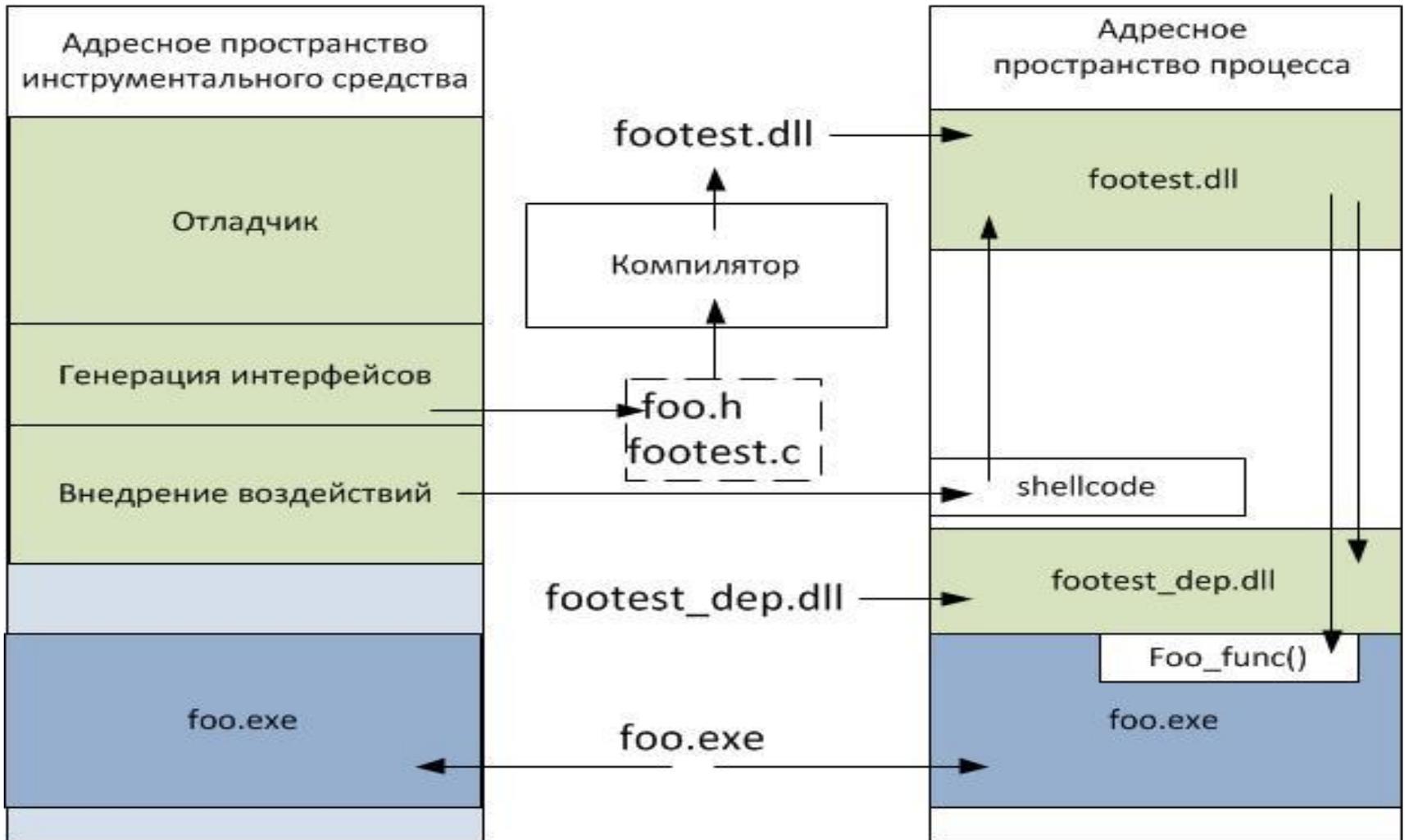
Фаззинг и модульные тесты

7



Система Dataflow

8



Откуда начинать?

9



Рейтинг функций

10

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

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

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

Матрица достижимости

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

12

Пример

Фаззер файлов .pdf для программы Evince

Тестовый запуск приложения

13

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

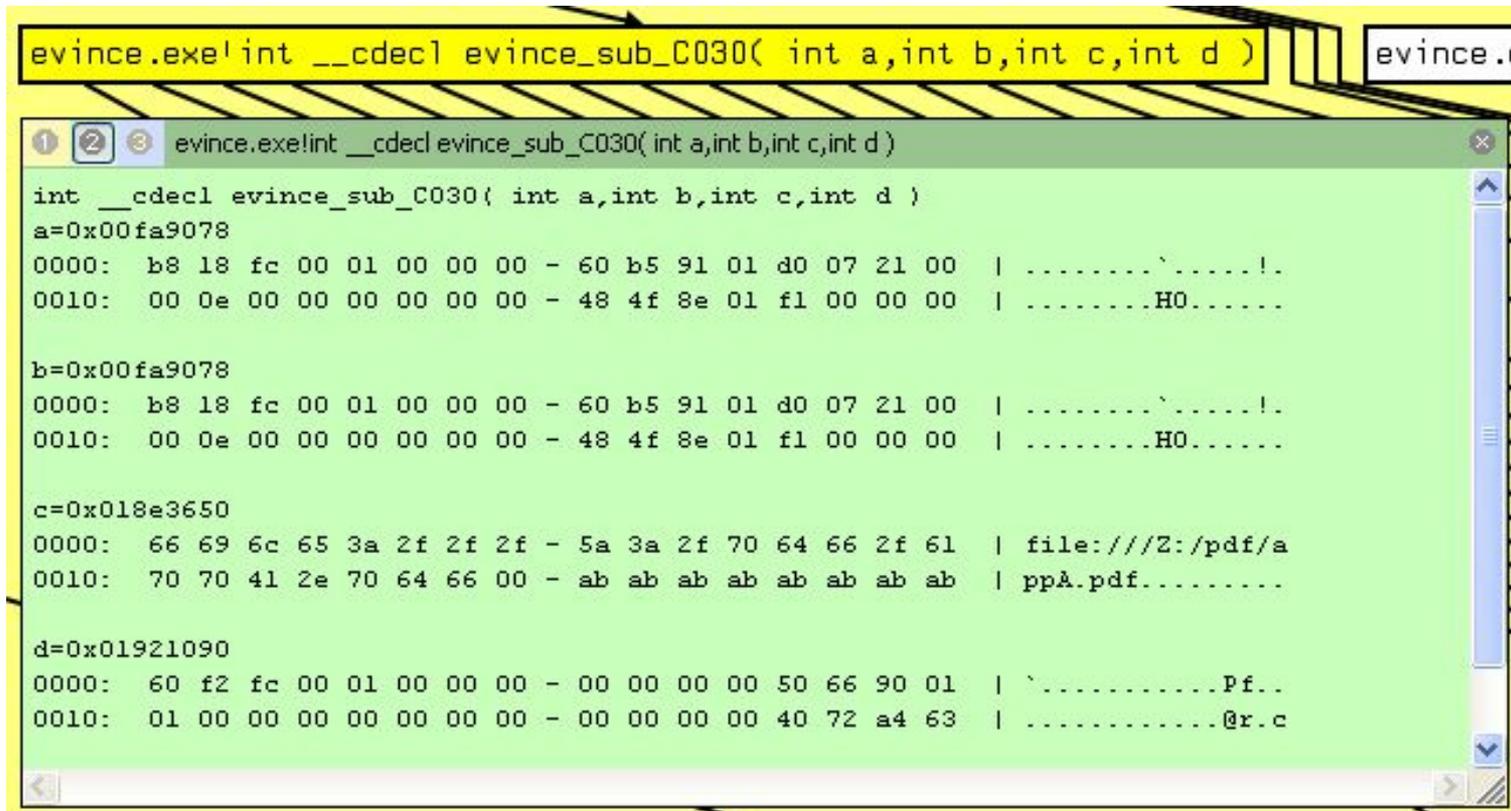


```
C:\Work\MaiWay\trunk\build\Debug\bin\dataflow.exe
DataFlow Debugger v0.2.0
Dataflow:DataflowServer, (-,127) XmlrpcServer::StartNetworkServer(): Server th
ad started
Process 84c is started
Module c:\program files\evince-2.30.3\bin\evince.exe loaded to base 0x400000
Module C:\WINDOWS\system32\ntdll.dll loaded to base 0x7c900000
Module C:\WINDOWS\system32\kernel32.dll loaded to base 0x7c800000
```

Функции, принимающие данные

14

Evince: 0xC030



```
evince.exe!int __cdecl evince_sub_C030( int a,int b,int c,int d ) evince.e
evince.exe!int __cdecl evince_sub_C030( int a,int b,int c,int d )
int __cdecl evince_sub_C030( int a,int b,int c,int d )
a=0x00fa9078
0000: b8 18 fc 00 01 00 00 00 - 60 b5 91 01 d0 07 21 00 | .....`.....!
0010: 00 0e 00 00 00 00 00 00 - 48 4f 8e 01 f1 00 00 00 | .....HO.....

b=0x00fa9078
0000: b8 18 fc 00 01 00 00 00 - 60 b5 91 01 d0 07 21 00 | .....`.....!
0010: 00 0e 00 00 00 00 00 00 - 48 4f 8e 01 f1 00 00 00 | .....HO.....

c=0x018e3650
0000: 66 69 6c 65 3a 2f 2f 2f - 5a 3a 2f 70 64 66 2f 61 | file:///Z:/pdf/a
0010: 70 70 41 2e 70 64 66 00 - ab ab ab ab ab ab ab ab | ppA.pdf.....

d=0x01921090
0000: 60 f2 fc 00 01 00 00 00 - 00 00 00 00 50 66 90 01 | `.....Pf..
0010: 01 00 00 00 00 00 00 00 - 00 00 00 00 40 72 a4 63 | .....@r.c
```

Функции, принимающие данные

15

Evince: 0x3290

```
evince.exe!int __stdcall evince_sub_3290( int a,int b,int c,int d,int e,int f )
evince.exe!int __stdcall evince_sub_3290( int a,int b,int c,int d,int e,int f )
int __stdcall evince_sub_3290( int a,int b,int c,int d,int e,int f )
a=0x00f45158
0000:  c8 8a f9 00 01 00 00 00 - 00 00 00 00 00 00 00 00 | .....
0010:  c0 9a f9 00 a0 9b f9 00 - 20 36 f4 00 18 50 f4 00 | ..... 6...P..

b=0x018e3650
0000:  66 69 6c 65 3a 2f 2f 2f - 5a 3a 2f 70 64 66 2f 61 | file:///Z:/pdf/a
0010:  70 70 41 2e 70 64 66 00 - ab ab ab ab ab ab ab ab | ppA.pdf.....

c=0x00f49038
0000:  40 ad f8 00 01 00 00 00 - e0 b7 91 01 00 00 00 00 | @.....
0010:  68 7e f4 00 00 00 00 00 - 00 00 00 00 00 00 00 00 | h~.....

d=0x00000000
e=0x00000000
```

Рейтинг функций

16

evince.exe!sub_C030(14854bcd7eaaa834773e90a42bbcc7e2)

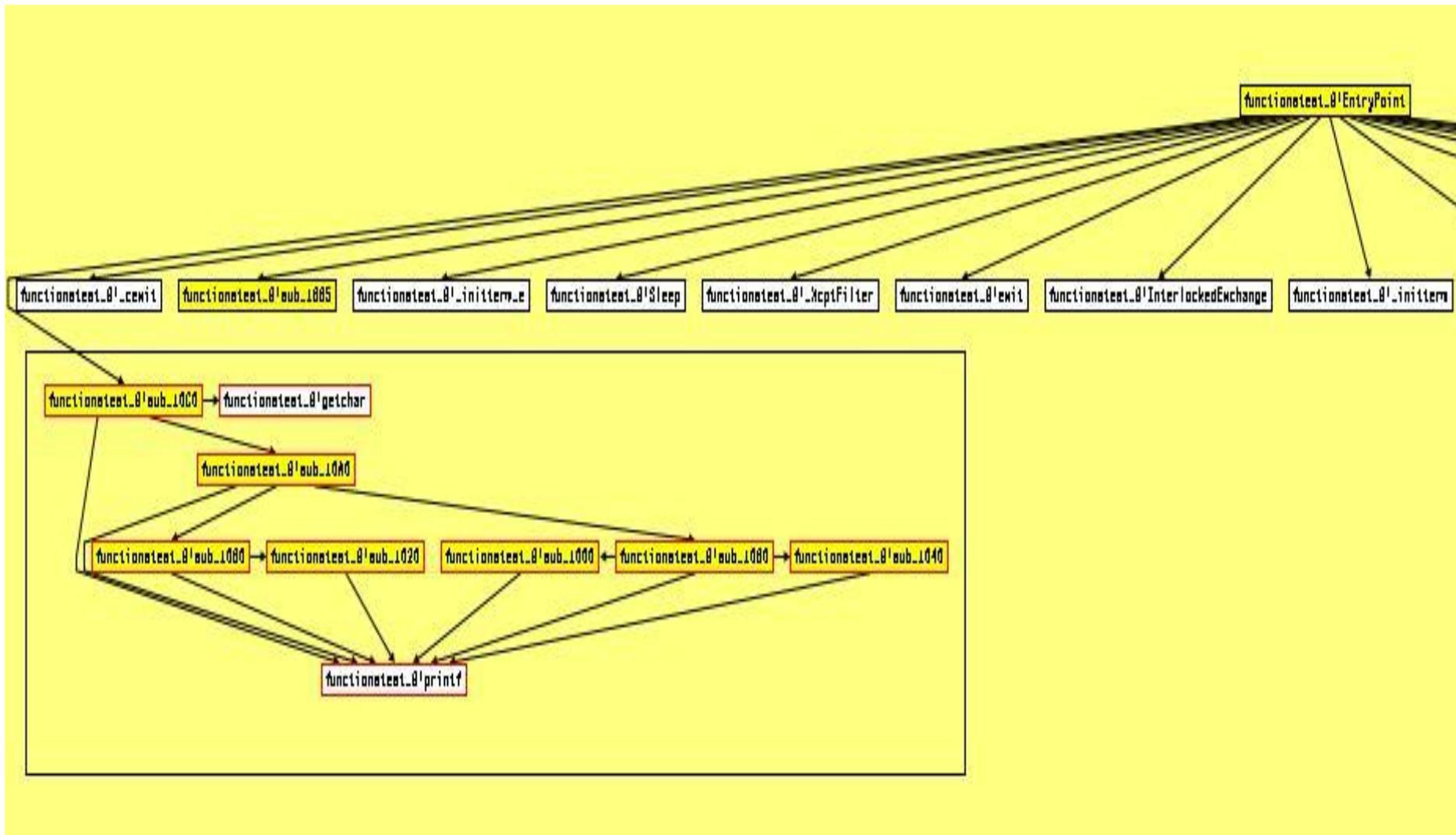
```
evince.exe!sub_C030(14854bcd7eaaa834773e90a42bbcc7e2)
path: c:\program files\evince-2.30.3\bin\evince.exe
Base: 400000
Module code size: 189540
Fuzzing potential reached code size: 166
Reached code size: 11580
Covered code size: 5476
```

evince.exe!sub_3290(14854bcd7eaaa834773e90a42bbcc7e2)

```
evince.exe!sub_3290(14854bcd7eaaa834773e90a42bbcc7e2)
path: c:\program files\evince-2.30.3\bin\evince.exe
Base: 400000
Module code size: 189540
Fuzzing potential reached code size: 9142
Reached code size: 11580
Covered code size: 5476
```

Оценка потенциального охвата

17



Подготовка теста.

Возможности

18

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

Подготовка теста

19

```
#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

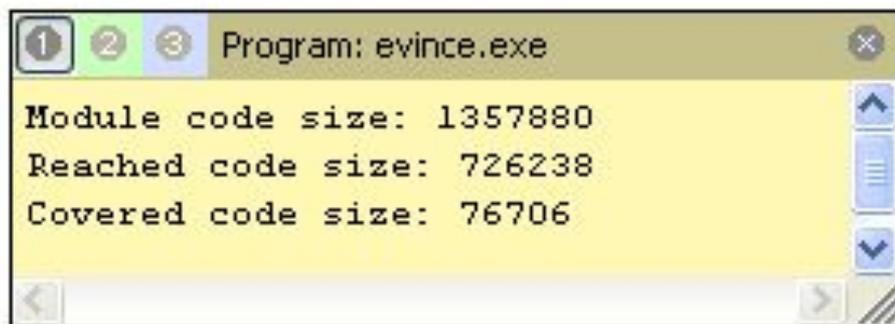
20

- 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

21

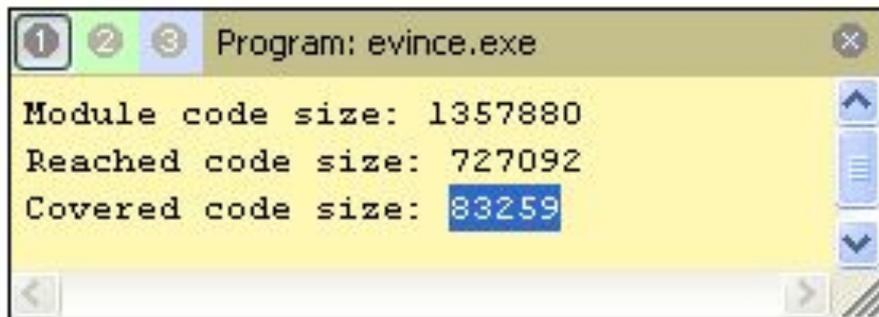
Program: evince.exe



1 2 3 Program: evince.exe

```
Module code size: 1357880
Reached code size: 726238
Covered code size: 76706
```

Program: evince.exe



1 2 3 Program: evince.exe

```
Module code size: 1357880
Reached code size: 727092
Covered code size: 83259
```

22

Вопросы?