

Архитектура и функциональные возможности инструмента ИСП Фаззер

Докладчик: Мишечкин Максим Владимирович
Акользин Виталий Владимирович
Курмангалеев Шамиль Фаимович

11.12.2020



Требования к современному фаззеру:

- Анализ программ через различные входы.
- Обратная связь по покрытию.
- Статическая и динамическая инструментации.
- Качественная фильтрация аварийных завершений.
- Интеграция с различными средствами динамического анализа.
- Отображение достигаемого покрытия.
- Удобное отображение найденных аварийных завершений.
- Фаззинга приложений принимающие на вход контейнеры.
- Фаззинг сетевых протоколов и сложных форматов данных.
- Возможность создания пользовательских мутаций.
- Интеграция с инструментами динамического символьного выполнения.
- Масштабируемость.
- Система управления многопоточного фаззинга.
- Наличие генератора входных данных на основе грамматик.
- Учет покрытия модуля и библиотек, используемых в программном обеспечении.

Сравнение фаззеров

	AFL	AFL++	AFL Smart	Peach	aggroArgs	ИСП Фаззер
Обратная связь по покрытию	+	+	+	-	-	+
Фаззинг файлов	+	+	+	+	-	+
Фаззинг сети	-	-	-	+	-	+
Фаззинг аргументов командной строки	-	-	-	-	+	+
Фаззинг stdin	+	+	+	-	-	+
Отображение покрытого бинарного кода	-	-	-	-	-	+
Описание формата данных(в т.ч. Peach)	-	-	+	+	-	+
Фаззинг контейнеров	-	-	-	-	-	+
Генератор на основе грамматик	-	-	-	-	-	+

Сравнение фаззеров

	AFL	AFL++	AFLSmart	Peach	aggroArgs	ИСП Фаззер
Интеграция с средствами динамического анализа	-	+	-	-	-	+
Пользовательские мутации	-	+	-	+	-	+
Символьное выполнение	-	-	-	-	-	+
Система управления многопоточного фаззинга	-	-	-	-	-	+
Качественная фильтрация аварийных завершений	-	-	-	-	-	+
Масштабируемость	+/-	+/-	+/-	-	-	+
Планировщики выбора данных на мутацию	-	+	-	-	-	-

Архитектура системы управления ИСП Фаззер

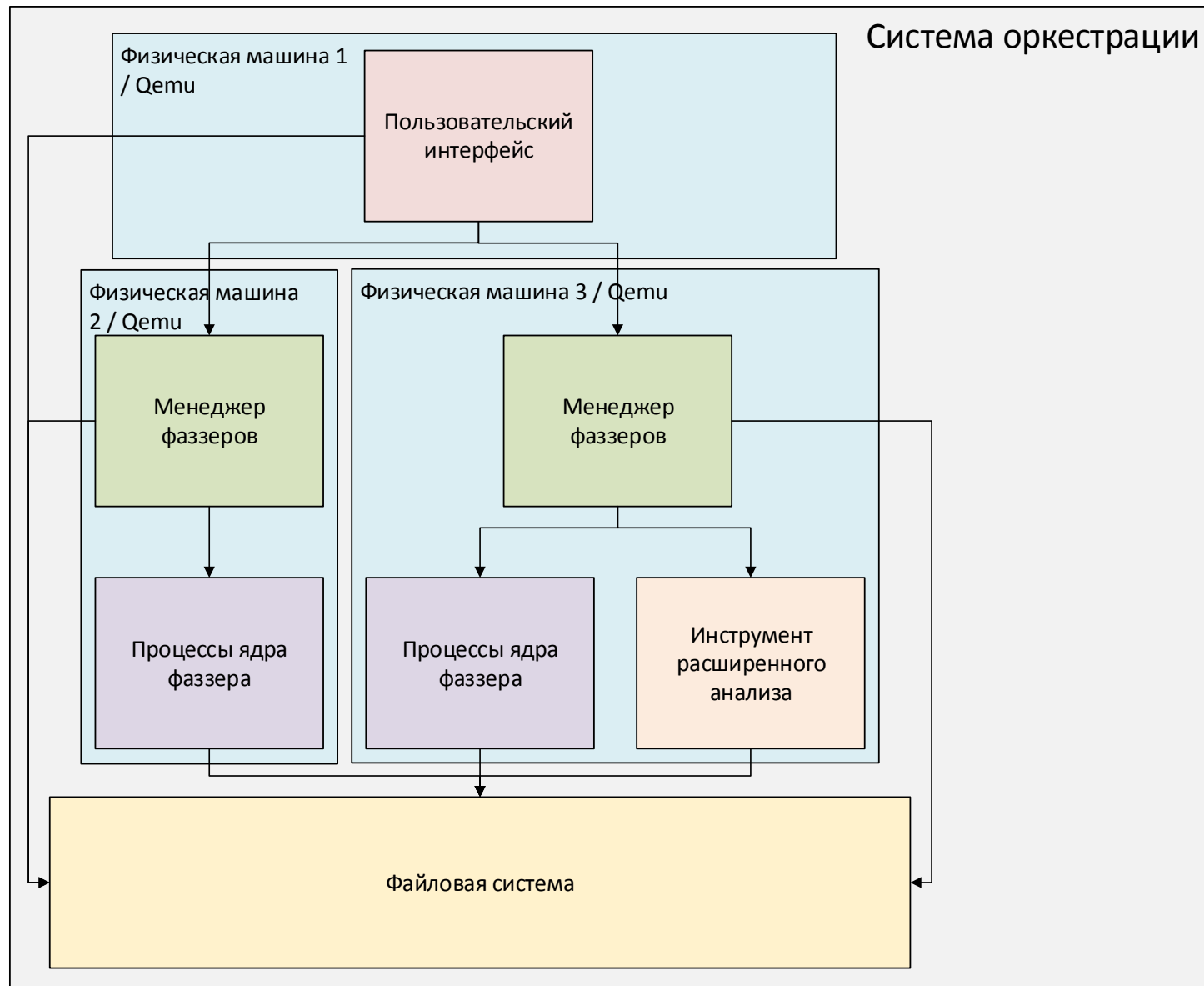
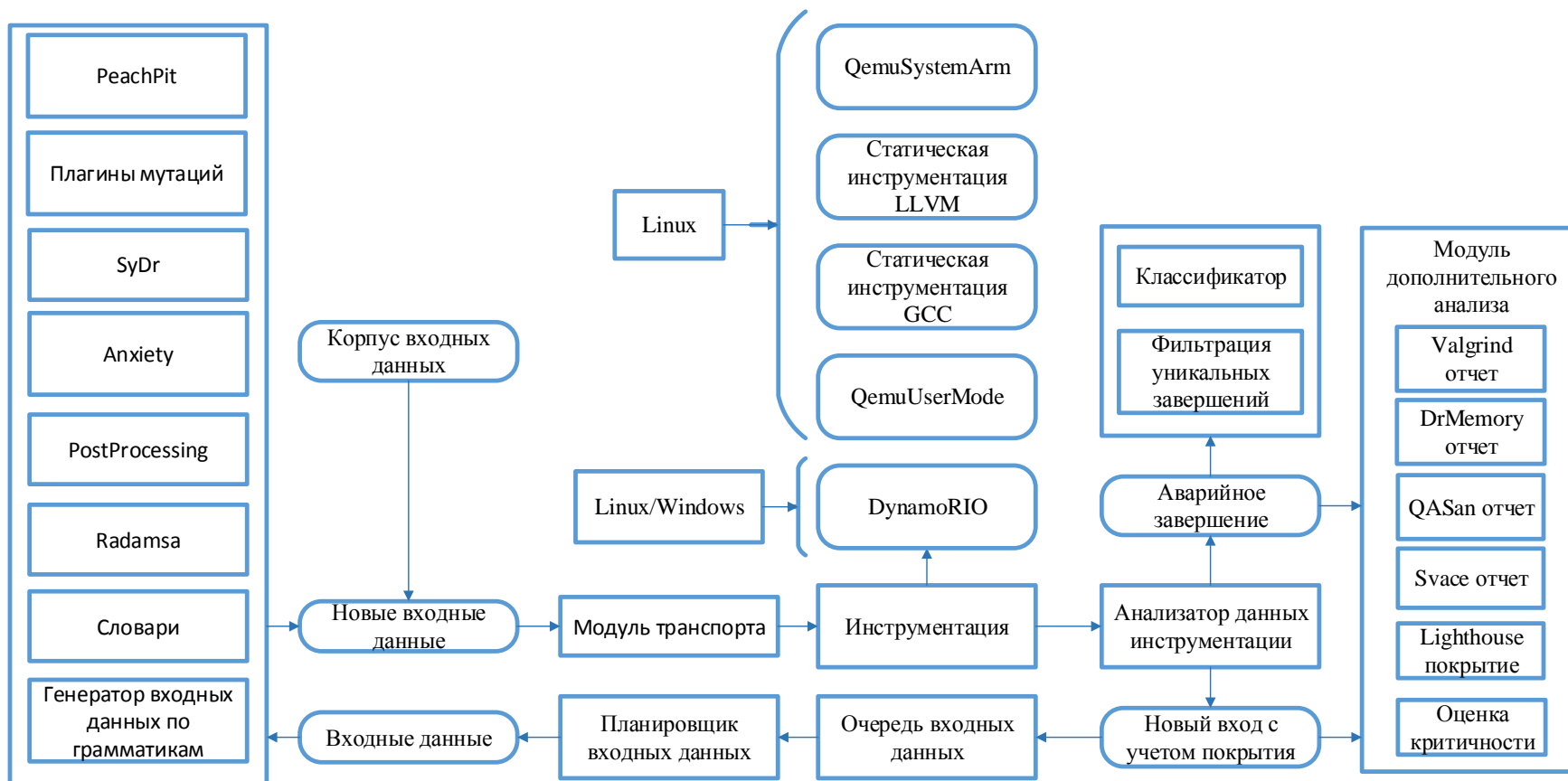


Схема ИСП Фаззера



Инструментации ИСП Фаззер

- Статическая инструментация :
 - GCC/G++
 - LLVM mode
- Динамическая инструментация:
 - QemuUserMode
 - DynamoRio
 - На основе LuaQemu
 - QemuSystemARM
 - QemuSystemX86
 - QemuSystemX64
 - ...

Функциональные возможности

- Анализ драйверов и приложений операционных систем Windows/Linux, встраиваемого ПО («прошивок») с использованием полносистемного эмулятора на основе LuaQemu.
- Анализ сетевых протоколов с использованием Reach описаний формата и пользовательских скриптов отправки данных.
- Возможность пост и пред обработки для анализа ПО принимающего на вход контейнеры и сложные форматы данных.

Инструмент расширенного анализа

- Покрытие по базовым блокам в формате Lighthouse.
- Отчеты Valgrind.
- Отчеты DrMemory.
- Отчеты QSan.
- Отчет аварийного завершения в формате svres, который можно загрузить в проект Svace.

Пример Lighthouse покрытия

```
sub     edx, ecx
add     ecx, eax
xor     edx, r8d
a = rcx ; uint32_t
sub     eax, edx
rorx   r9d, edx, 1Ch
add     edx, ecx
xor     eax, r8d
length = rsi ; size_t
b = rdx ; uint32_t
k = rdi ; const uint16_t *
cmp    length, 0Ch
ja     def_25928E ; jumtable 000000000025928E default case
```



покрытая часть кода



непокрытая часть кода

```
loc_259280:
lea    r8, jpt_25928E
movsxd length, ds:(jpt_25928E - 5037C0h)[r8+length*4]
add    rsi, r8
jmp    rsi ; switch jump
```

```
loc_259531: ; jumtable 000000000025928E case 23
movzx  esi, byte ptr [k+0Ah]
shl    esi, 10h
add    eax, esi
```

```
; jumtable 000000000025928E case 21
k+8]
0000000h]
```

```
loc_25953A: ; jumtable 000000000025928E case 22
movzx  esi, word ptr [k+8]
add    eax, esi
jmp    short loc_259520 ; jumtable 000000000025928E case 20
```

```
loc_259573: ; jumtable 000000000025928E case 24
movzx  esi, word ptr [k+8]
add    eax, esi
movzx  esi, word ptr [k+0Ah]
shl    esi, 10h
add    eax, esi
c = rax ; uint32_t
jmp    short loc_259520 ; jumtable 000000000025928E case 20
; } // starts at 259180
hashlittle_safe endp
```

```
loc_259562: ; jumtable 000000000025928E case 19
movzx  esi, byte ptr [k+6]
shl    esi, 10h
add    edx, esi
```

```
loc_259520: ; jumtable 000000000025928E case 20
movzx  esi, word ptr [k+4]
add    edx, esi
movzx  esi, word ptr [k+6]
shl    esi, 10h
add    edx, esi
b = rdx ; uint32_t
jmp    short loc_259500 ; jumtable 000000000025928E case 16
```

```
loc_259568: ; jumtable 000000000025928E case 18
movzx  esi, word ptr [k+4]
add    edx, esi
jmp    short loc_259500 ; jumtable 000000000025928E case 16
```

```
loc_25954F: ; jumtable 000000000025928E case 15
k = rdi ; const uint16_t *
movzx  esi, byte ptr [k+2]
shl    esi, 10h
add    ecx, esi
```

Отчет аварийного завершения в Svace

Svace warnings x +

localhost:8060/history/svacelight/index.html#show/test1/master/43138056b03d188970cbe1e458e3fd93e5882dd8/dd

test1 master Fri Dec 28 16:51:09 MSK 2018 Show .svres:exp

WARNING_BY_FUZZER (6)

WARNING_BY_FUZZER (crashed-4) Program can crash here with signal 11 new
prog.c:10

WARNING_BY_FUZZER (crashed-2) Program can crash here with signal 11 new
prog.c:10

WARNING_BY_FUZZER (crashed-0) Program can crash here with signal 11 new
prog.c:10

WARNING_BY_FUZZER (crashed-1) Program can crash here with signal 11 new
prog.c:5

- [backtrace] crash1 at prog.c:5
- fun at prog.c:25
- main at prog.c:34

Add comment

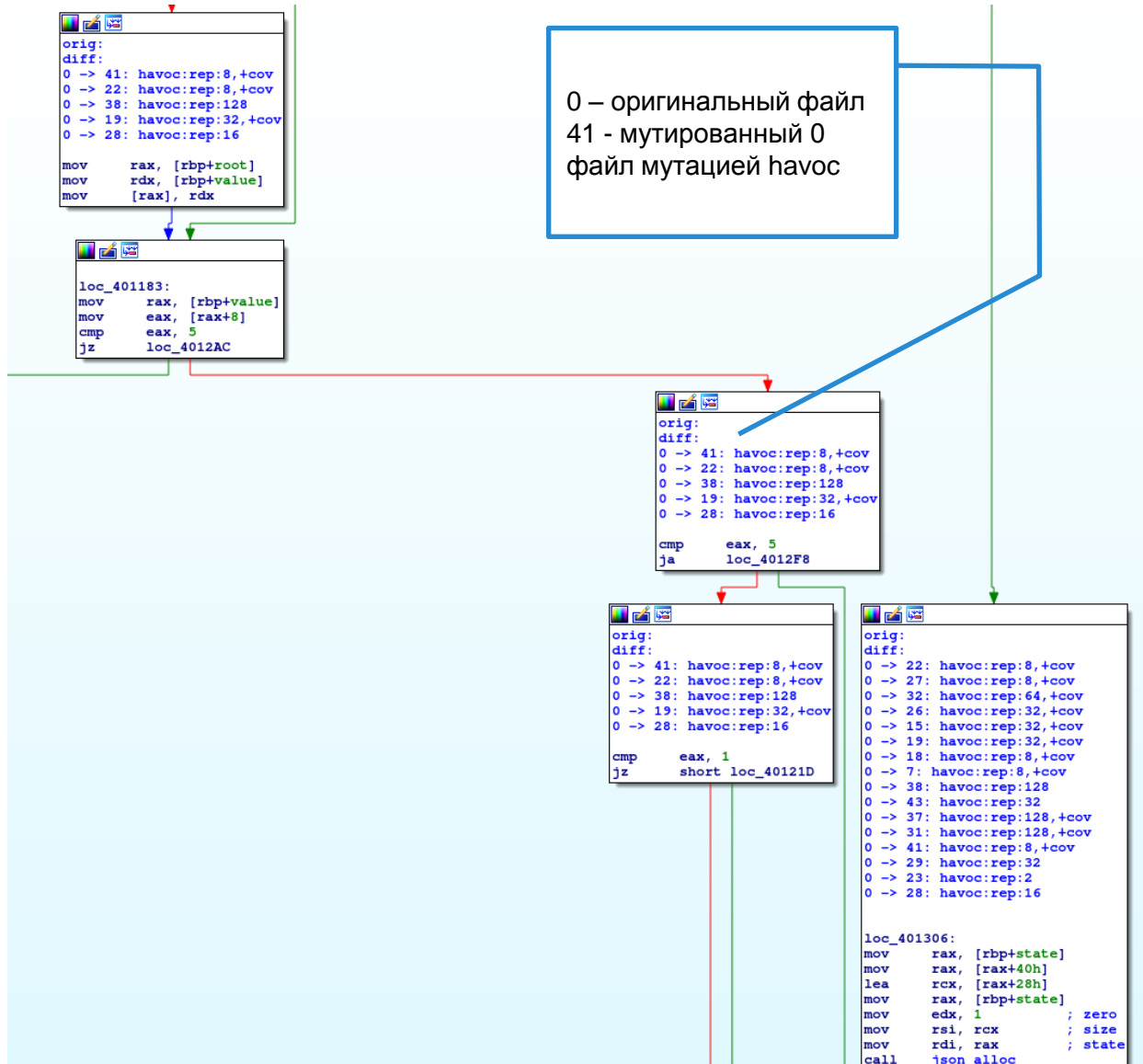
History Undecided Unspecified

WARNING_BY_FUZZER (crashed-3) Program can crash here with signal 11 new

/home/fedor/fuzzer/svace-test/test1/prog.c

```
1 #include <stdio.h>
2
3 void crash1(void)
4 {
5     *(int*)0 = 1;
6 }
7
8 void crash2(void)
9 {
10    *(int*)0 = 2;
11 }
12
13 void fun(int c)
14 {
15     int s = 0;
16     int i;
17     for (i = 0; i < c % 5 + 4; i++) {
18         s += i;
19     }
20     if (c % 2 == 0)
21         s += 1;
22     if (c % 3 == 0)
```

Технология MIA



Применение ИСП Фаззера

Результаты фаззинга popler Release 20.12.1 :

- Число путей: 4571
- Утечек памяти: 2
- Использование неинициализированной переменной: 4

Результаты фаззинга mupdf-1.18.0:

- Переполнение буфера на куче: 1

Результаты фаззинга wavpack 5.3.2:

- Найдено 10 уникальных аварийных завершений.

Применение ИСП Фаззера

Фаззинг DXE драйверов UEFI BIOS без наличия исходного кода методами:

- Частичной эмуляции LuaQemu.
- Пересборки DXE драйверов PE32+ формата в ELF с заменой UEFI специфичного кода.

Обнаруженные дефекты:

- Разыменование нулевого указателя.
- Деление на ноль.
- Выход драйвера из строя(DoS).
- Переполнение буфера на куче.

Результаты были получены с использованием:

- Покрытия в формате Lighthouse.
- Символьного выполнения Sydr.
- Отчетов Valgrind.



Спасибо за внимание



Мишечкин Максим
mish.max@ispras.ru