ИСПРАН

# Анализ бинарного кода программ

Вартан Андроникович Падарян
vartan@ispras.ru

# Цели и предметы анализа

- Исследование свойств стороннего ПО в отсутствии исходного кода
  - Контроль отсутствия НДВ
  - Выявление программных дефектов
  - Оценка влияния программного дефекта на безопасность ПО
  - Восстановление сетевых протоколов
- Мониторинг поведения эксплуатирующейся системы
  - Выявление уязвимостей нулевого дня
  - Соблюдение политик безопасности
- Анализ сетевого трафика
  - Выявление сетевых атак в трафике сверхвысокого объема

Приложения

Ядро ОС и драйвера

Загрузчики и гипервизоры

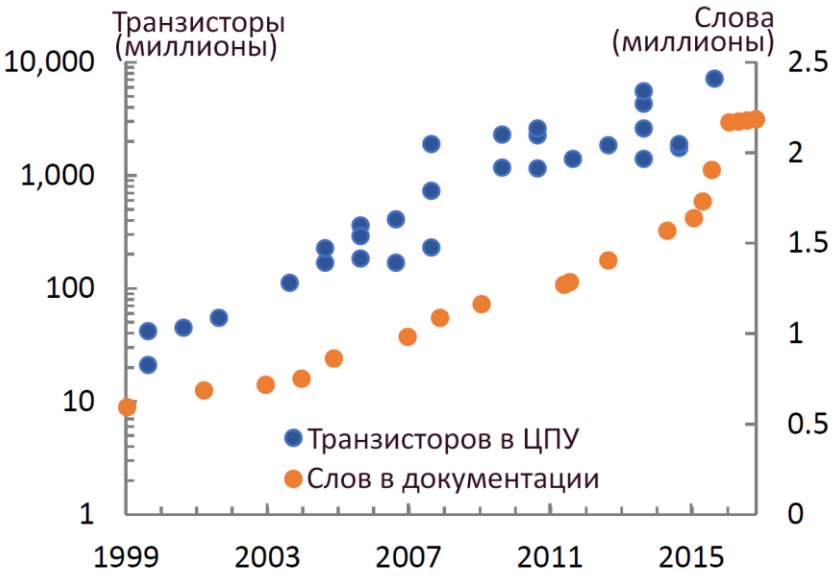
Конечные устройства: настольные компьютеры, сервера, смартфоны/планшеты, IoT, ...

Оборудование локальных сетей: маршрутизаторы, межсетевые экраны, беспроводная связь, ...

Сетевой трафик в локальных сетях и на магистральных каналах

# Условия проведения анализа

- Главное требование к средствам анализа
  - *Разрабатывается единожды – применяется везде*
- Значительный объем анализируемого кода
  - Встроенное ПО – порядка нескольких МБ
  - Типичное приложение с библиотеками – порядка сотен МБ
- Исполняемый код строится оптимизирующими компиляторами, применяется запутывание кода
- Непрерывное усложнение системы команд архитектуры x86
  - 23 расширения команд в период 2011-2016
  - Значительная часть – системные команды
    - VMFUNC, SGX, MPK, CET
- IoT
  - Множество различных SoC и процессорных архитектур




*Andrew Baumann (Microsoft Research). Hardware is the new software. // 16th Workshop on Hot Topics in Operating Systems, May 2017*

# Традиционные методы и инструменты анализа

Необходимо одновременно видеть код и «типовые» данные, которые он обрабатывает

- Статический анализ
  - Дизассемблер/ Декомпилятор (IDA Pro/HexRays)
  - Среда анализа (IDA Pro + дополнения, CodeSurfer/x86)
- Динамический анализ
  - Обычный отладчик, Отладчик ядра (SoftICE, OllyDbg, WinDbg)
  - Отладчик на основе виртуальной машины
  - Трассировщик (Post-Mortem vs. Online)
- Исследовательские проекты
  - Методы и инструменты анализа BitBlaze, S2E, Mayhem, Radare2, Triton, Panda, …

```
[0x00018b30 65% 555 /bin/sh]> pd $r @ entry0
      |||   ;-- entry0:
      |||   0x00018b30      31ed           xor ebp, ebp
      |||   0x00018b32      4989d1         mov r9, rdx
      |||   0x00018b35      5e             pop rsi
      |||   0x00018b36      4889e2         mov rdx, rsp
      |||   0x00018b39      4883e4f0       and rsp, 0xfffffffffffffff0
      |||   0x00018b3d      50             push rax
      |||   0x00018b3e      54             push rsp
      |||   0x00018b3f      4c8d05ea3b00.  lea r8, 0x0001c730          ; 0x1c730
      |||   0x00018b46      488d0d733b00.  lea rcx, 0x0001c6c0         ; 0x1c6c0
      |||   0x00018b4d      488d3d0c5eff.  lea rdi, 0x0000e960         ; 0xe960
      |||   0x00018b54      e847bafeff     call sym.imp.__libc_start_main ;[1]
      |||   0x00018b59      f4             hlt
      |||   0x00018b5a      660f1f440000   nop word [rax + rax]
      |||   0x00018b60      488d3db9c620.  lea rdi, obj._environ       ; 0x225220 ; obj.__environ ; obj.__envir
      |||   0x00018b67      488d05b9c620.  lea rax, 0x00225227         ; 0x225227 ; "tab" ; "tab" 0x00225227
      |||   0x00018b6e      55             push rbp
      |||   0x00018b6f      4829f8         sub rax, rdi
      |||   0x00018b72      4889e5         mov rbp, rsp
      |||   0x00018b75      4883f80e       cmp rax, 0xe
     ,====< 0x00018b79      7615           jbe 0x18b90                 ;[2]
     ||||   0x00018b7b      488b05cec120.  mov rax, qword [reloc._ITM_deregisterTMCloneTable_80] ; [0x224d50:8]
     ||||   0x00018b82      4885c0         test rax, rax
    ,=====< 0x00018b85      7409           je 0x18b90                  ;[2]
    |||||   0x00018b87      5d             pop rbp
    |||||   0x00018b88      ffe0           jmp rax
    |||||   0x00018b8a      660f1f440000   nop word [
    ``----> 0x00018b90      5d             pop rbp
      |||   0x00018b91      c3             ret
      |||   0x00018b92      0f1f4000       nop dword
      |||   0x00018b96      662e0f1f8400.  nop word c
    ..----> 0x00018ba0      488d3d79c620.  lea rdi, o                                                    vir
    |||||   0x00018ba7      488d3572c620.  lea rsi, o                                                    vir
    |||||   0x00018bae      55             push rbp
    |||||   0x00018baf      4829fe         sub rsi, r
    |||||   0x00018bb2      4889e5         mov rbp, r
    |||||   0x00018bb5      48c1fe03       sar rsi, 3
    |||||   0x00018bb9      4889f0         mov rax, r
    |||||   0x00018bbc      48c1e83f       shr rax, 0x3
    |||||   0x00018bc0      4801c6         add rsi, rax                ; '#'
    |||||   0x00018bc3      48d1fe         sar rsi, 1
   ,======< 0x00018bc6      7418           je 0x18be0                  ;[3]
   ||||||   0x00018bc8      488b05f9c320.  mov rax, qword [reloc._ITM_registerTMCloneTable_200] ; [0x224fc8:8]=
```

Radare2
Свободно распространяемый дизассемблер и фреймворк для реверс-инжиниринга
http://radare.org/

# BinNavi – расширение возможностей статического анализа

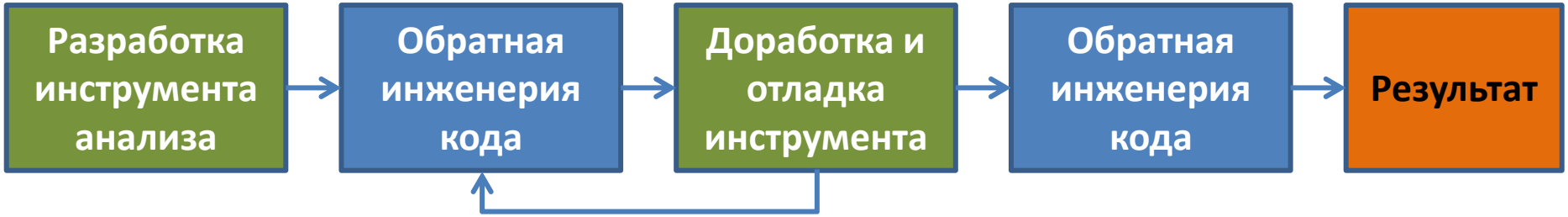
- Восстанавливается граф потока выполнения
- Восстанавливаются потоки данных
- Возможность построения слайсов
- Поддерживаются архитектуры x86, PPC32, ARM

- Анализ выполняется поверх внутреннего представления REIL
  - 17 команд

# «Нестандартный» пример

- Поведение определяется обрабатываемыми данными
  - Необходимы результаты динамического анализа
- Нестандартная операционная система
  - Наилучший случай: модифицированный Linux, BSD, …
- Доступна только прошивка, часть аппаратуры неизвестна
  - ***Запустить прошивку невозможно***

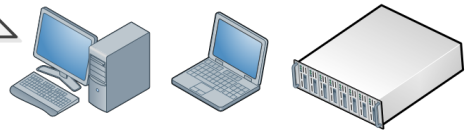
# Решаемые задачи



ИСП РАН

| | Анализ во время выполнения (online) | Анализ трасс выполнения (offline) | |
|---|---|---|---|
| Прикладные задачи | Сервер-приманка (honeypot) | Контроль отсутствия НДВ | Исследование вредоносного ПО |
| Анализ потоков данных и управления | Фиксация срабатывания ошибок | Извлечение спецификации алгоритма | Восстановление формата данных |
| Повышение уровня представления | Интроспекция виртуальной машины | Восстановление статико-динамического представления программы | |
| Контролируемая среда выполнения на базе эмулятора | Интерфейс отладчика | Сбор трасс | |
| | Детерминированное воспроизведение | Механизм «обратной» отладки | |
| Разработка новых виртуальных платформ | Процессор | Периферийные устройства | Интеграция виртуальной платформы |

Анализ ПО устройств различных классов



Персональные компьютеры и сервера

Мобильные устройства

Коммуникационное оборудование

Встраиваемые системы и загрузчики

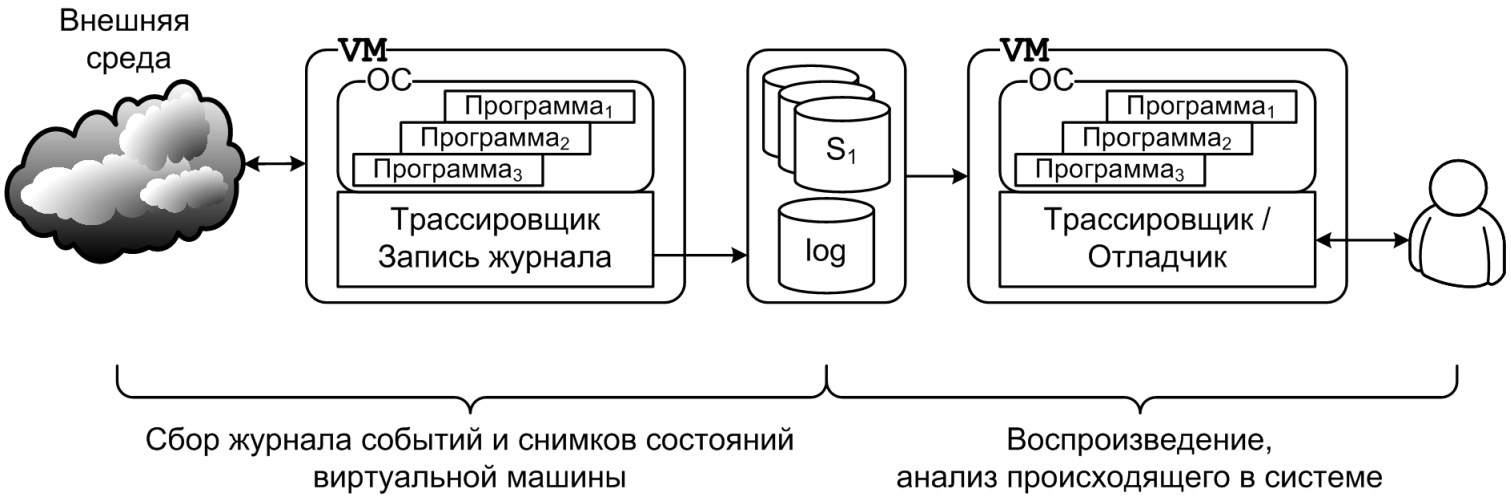
# Зачем нужны программные эмуляторы?

- Консолидация серверов – традиционное применение
- Полносистемный эмулятор в разработке ПО
  - запуск приложений для «редких» архитектур
  - инструмент разработчика
  - инструмент аналитика в вопросах ИБ
- Аудит безопасности или обратная инженерия ПО
  - необходимо иметь возможность выполнить ПО в контролируемой среде
  - желательно иметь возможно записать подробный ход выполнения программы
  - полезно иметь возможность обратной отладки

# Qemu – платформа динамического анализа

| | | | | |
|---|---|---|---|---|
| Инструменты для решения практических задач | Honeypot | Расширяемый отладчик уровня ОС | Поведенческий анализ | Контролируемая пользовательская среда |
| Технологии анализа | Анализ помеченных данных | Механизм отладки | Трассировка | |
| Расширяемость средств анализа | Механизм плагинов анализа в эмуляторе | | | |
| Базовые средства эмуляции | Эмулятор QEMU с механизмом детерминированного воспроизведения | | | |
| Расширяемость набора целевых ВМ | Автоматизация создания новых виртуальных машин | | | |
| | Процессор | Интеграция компонент виртуальной машины | Периферийные устройства | |

# Детерминированное воспроизведение работы виртуальной машины




- Проблема: «тяжеловесный» анализ (отладка, пошаговая трассировка и т.п.) приводит к замедлению работы ВМ на несколько порядков, что влияет на поведение исследуемого кода, если он взаимодействует с «внешним миром»
- Детерминированное воспроизведение гарантирует повторение выполнения с точностью до отдельной машинной команды
- Реализовано в ведущих коммерческих эмуляторах: SimNow, Simics, Synopsys Virtual Platform★
- В СПО эмулятор Qemu (начиная с версии 2.5, полная поддержка – версия 2.8) включен механизм воспроизведения, разработанный в ИСП РАН

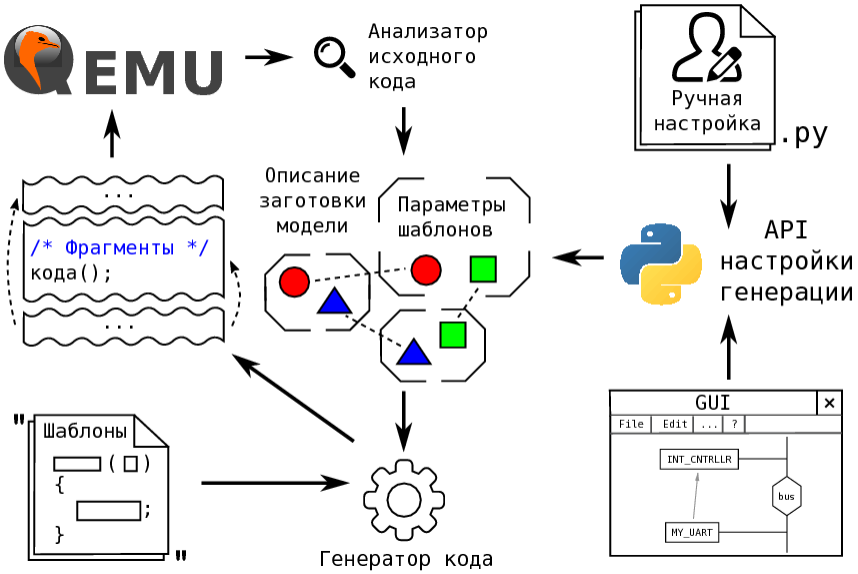


★Воспроизведение на уровне RTL

# Автоматизация разработки новой виртуальной машины в эмуляторе Qemu

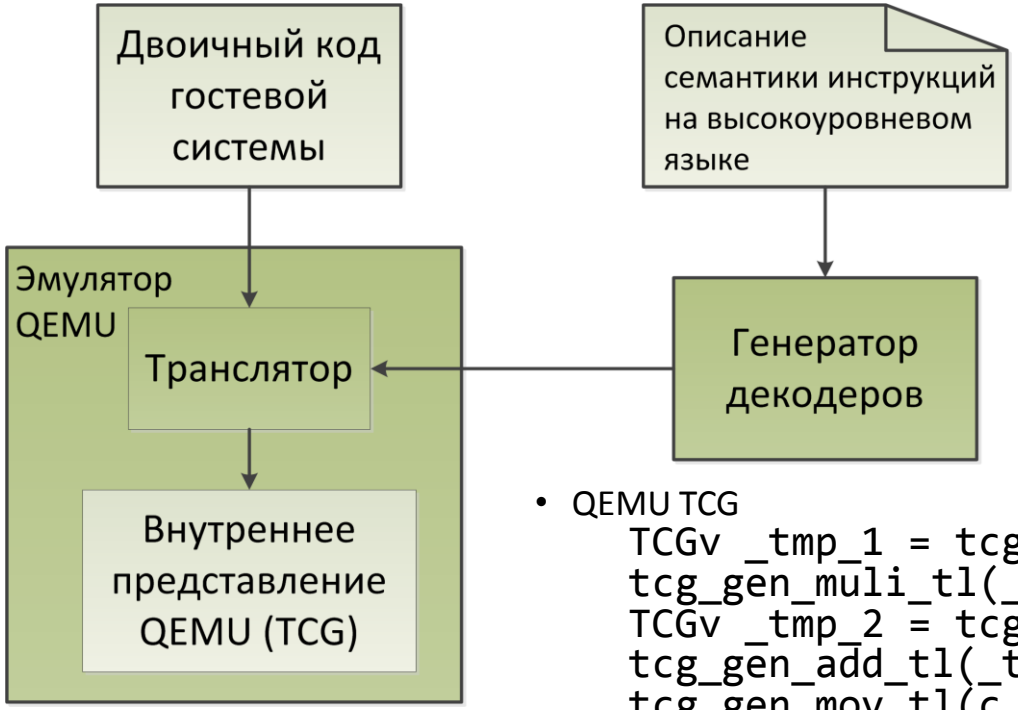
Создание новой виртуальной машины

- Поддержка процессорной архитектуры
- Поддержка новых периферийных устройств
- Интеграция компонент виртуальной машины




- Определены типовые виртуальные устройства и их атрибуты, конкретизирующиеся при реализации в эмуляторе Qemu.
- Разработан метод создания новых виртуальных устройств
  - Раннее выявление ошибок интеграции компонент
  - Возможность применять графические нотации

# Автоматизация поддержки нового процессора в эмуляторе Qemu




- QEMU TCG

```
TCGv _tmp_1 = tcg_temp_new();
tcg_gen_muli_tl(_tmp_1, a, 2);
TCGv _tmp_2 = tcg_temp_new();
tcg_gen_add_tl(_tmp_2, _tmp_1, b);
tcg_gen_mov_tl(c, _tmp_2);
tcg_temp_free(_tmp_1);
tcg_temp_free(_tmp_2);
```

- Pivot

```
r:c = add.i32(mul.i32((i32)2, r:a), r:b)
```

- nML

```
c = (a * 2) + b;
```

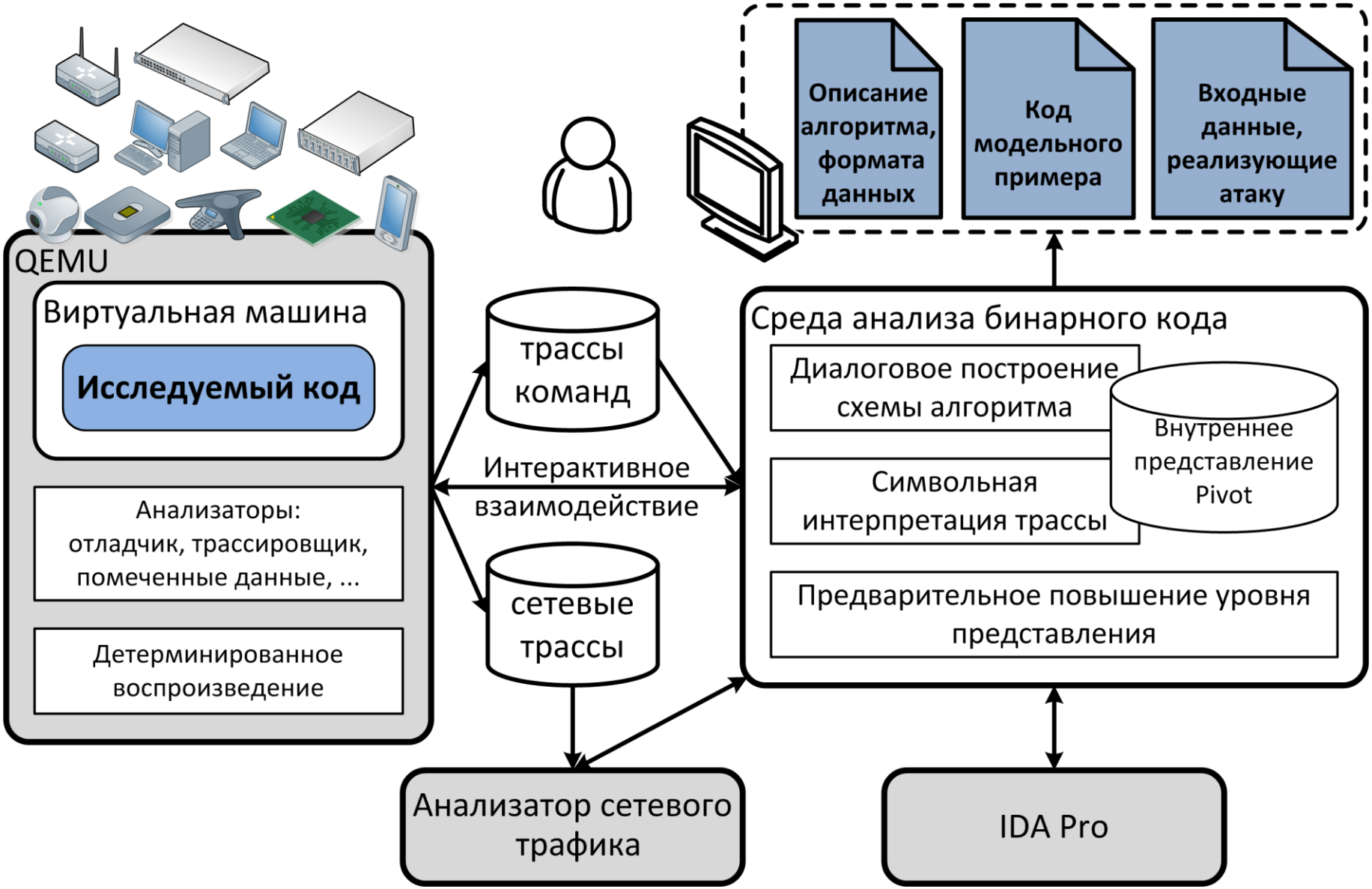
- Разработанное решение

```
c = (a * 2) + b;
```

# 2 места в команде разработчиков эмуляторов

- Для выполнения кода в эмуляторе
  - нужна поддержка необходимой процессорной архитектуры
  - нужна поддержка необходимого набора периферийных устройств
  - нужна возможно объединить все это в виртуальную вычислительную систему
- Задача: автоматизация процесса разработки эмулятора

- Для записи хода выполнения программы необходимо
  - детерминированное повторное воспроизведение
- Задача: разработка эффективного детерминированного повторного воспроизведения


Описание алгоритма, формата данных
Код модельного примера
Входные данные, реализующие атаку
QEMU
Виртуальная машина
Исследуемый код
Анализаторы: отладчик, трассировщик, помеченные данные, ...
Детерминированное воспроизведение
трассы команд
Интерактивное взаимодействие
сетевые трассы
Среда анализа бинарного кода
Диалоговое построение схемы алгоритма
Символьная интерпретация трассы
Внутреннее представление Pivot
Предварительное повышение уровня представления
Анализатор сетевого трафика
IDA Pro

# Поднятие уровня представления

«Мета PDS»: слияние кода из разных трасс, перебазирование адресов на начала модулей, замыкание ребер

Восстановление функций

PDS (Pushdown System)
Статическое представление кода одной трассы / Поколения кода

Распознавание модулей

Построение стеков вызовов

Поколения кода в памяти

Определение границ обработчиков прерываний

Разметка переключений контекста

```
LDRH    R1, [R4, 0x00000004]  ; [4
BEQ     0x407575B8  ; -> 407575B8
RSB     R2, R2, 0x00000005
ADD     PC, PC, R2, LSL #4  ; -> 40757588
AND     R12, R1, 0x00000F00
LDR     R2, [R5, R12, LSR #6]  ; [4205F588]
```

Трасса

```
pc = 40757554      r0 = 42174ae8
r4 = 43953bec      r5 = 4205f54c
r9 = 40d86950     r10 = 4205f538
r14 = 407500ac r13_svc = d6143ff8
r13_und = c0322498 r14_und = c0027a40
```

# Зависимости по данным на уровне машинных команд

Instruction Info

CALL 80564B80h

**General**

| | |
|---|---|
| Flags: | ControlTransfer \| ControlTransferCall |
| Explicit operand count: | 1 |
| Execution flow: | normal |
| Delay slot count: | 0 |

**Interrupt**

| | |
|---|---|
| Interrupt: | none |

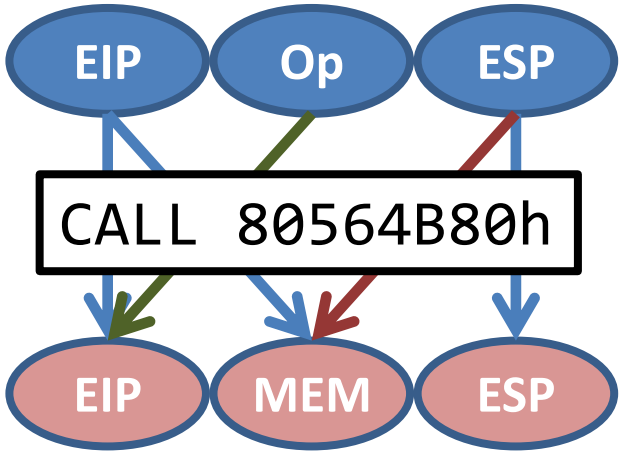
**Control Transfer**

| | |
|---|---|
| Expression: | |
| Target address: | 80564B80 |

```
1. local(0xFFFFFFFFFFFFFFE0, 0x4) = EIP
2. v(0xF7926C90, 0x4) = Set({ local(0xFFFFFFFFFFFFFFE0, 0x4), SS, ESP
   })
3. ESP = ESP
4. RIP = { RIP, v(0x80564C24, 0x4) }
```

☑ A ☑ C ☑ F ☑ I ☑ S ☑ X

Dependencies | Interrupt Dependencies | Pivot




- В одной машинной команде может выполняться несколько присвоений
- «Дробление» зависимостей между входными и выходными операндами уточняет анализ потоков данных

# Tx: Pivot

## TEST EAX, EAX

```
1.   INIT o.0:i16  = 0000h  # EAX
2.   LOAD t.0:i32  = r[o.0]
3.   APPL t.1      = and.i32(t.0, t.0)[:ACOPSZ]
4.   INIT t.2:i16  = 0088h  # EFLAGS
5.   LOAD t.3:i16  = r[t.2]
6.   INIT t.4:i16  = 08C5h
7.   APPL t.5      = x86.uf(t.3, t.4)[ACOPSZ:]
8.   STOR r[t.2]   = t.5
```

# 1 место (2 варианта темы) в команде Pivot 2