Скрытая отладка виртуальных машин

Довгалюк П. М., Абакумов М. А., Полетаев Д. Н., Иванов А. В., Степанов В. М.
Новгородский государственный университет
Институт системного программирования РАН

Отладка на удалённом компьютере

- Инструментальный код выполняется внутри виртуальной машины
  - Система должна функционировать
- Нельзя отладить любой выполняемый код
  - Загрузочный код
  - BIOS
- Нужно настраивать заранее
- Влияет на работу отлаживаемого кода

QEMU

- Отладочные возможности
  - Подключение удаленного отладчика gdb
  - Трассировка транслируемых и выполняемых блоков кода
- Симулятор с открытым исходным кодом
  - Можно моделировать и отлаживать собственные периферийные устройства или аппаратные платформы
- Виртуальные машины на платформах i386, ARM, MIPS, PowerPC и т.д.
  - 20+ семейств процессоров

Отладка ОС через gdbserver в симуляторе

- Можно подключиться в любой момент
  - Даже при критическом сбое
  - Даже до загрузки ОС
- Отладка без реального оборудования
  - Отладка прошивки/драйвера
  - Отладка моделей оборудования
- Модели оборудования не всегда идеальны
- Работает медленнее из-за виртуализации или ход работы может измениться из-за остановок
- Неполная информация о системе

1 Недочеты в эмуляции

- Ошибки в программной реализации
- Недокументированное поведение реального оборудования
- Временные характеристики эмуляции
- Порты или регистры для взаимодействия с гипервизором

2 Замедление/прерывание работы

- Виртуальное время
- Воспроизведение заранее записанных сценариев
  - Разделение записи и анализа
  - Обратная отладка через gdb

Запись и воспроизведение

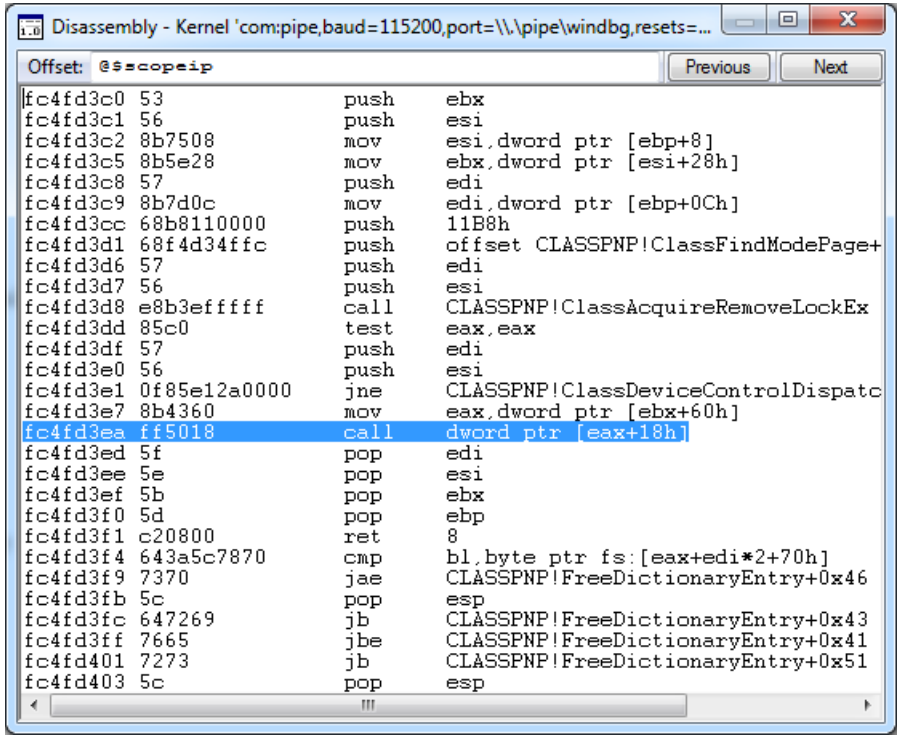
- Отладка и анализ всей системы
- Протестированы для x86, x86-64, ARM, MIPS
- Нет воздействия на гостевую систему при анализе
  - Профилирование
  - Анализ помеченных данных
  - Отладка
  - Трассировка

3 Неполная информация о системе

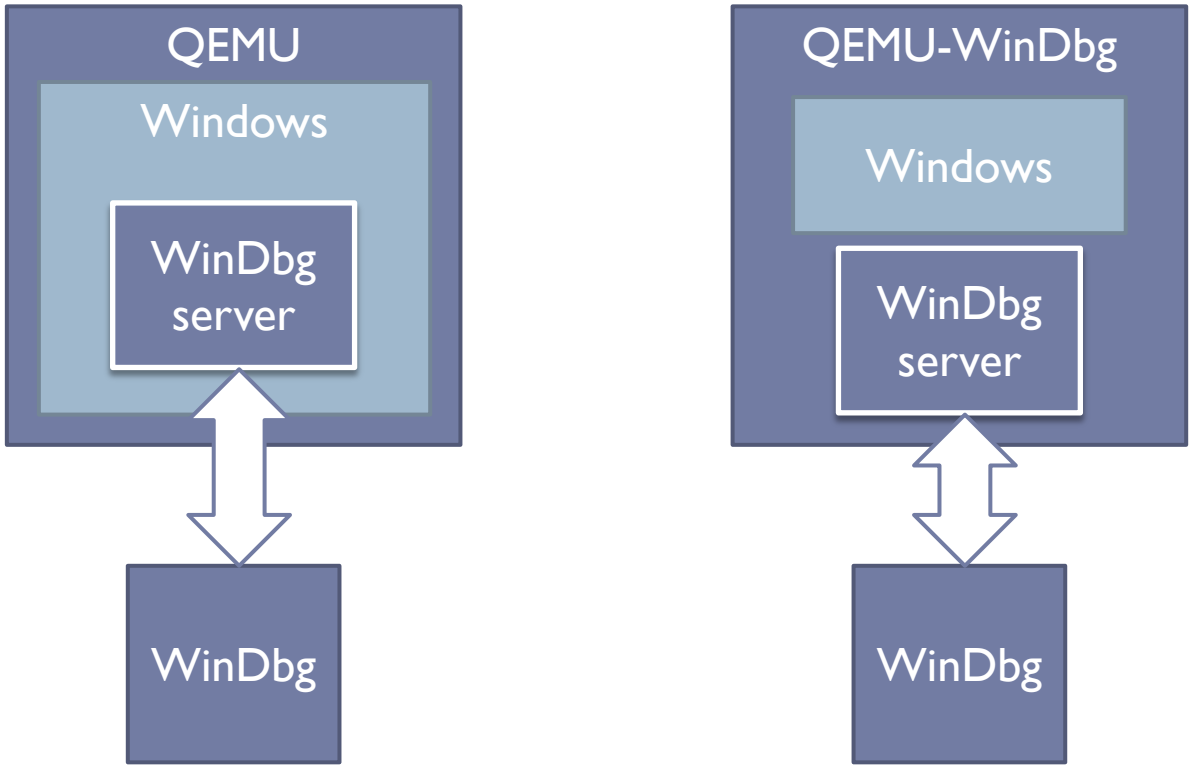
- Не видно потоков и процессов
- Нет символьной информации, если это не Linux

```
   0x77dde081:  call   *0x77dd11fc
   0x77dde087:  mov    %eax,%ebx
   0x77dde089:  test   %ebx,%ebx
   0x77dde08b:  jl     0x77dde097
   0x77dde08d:  test   %ebx,%ebx
   0x77dde08f:  jl     0x77dde097
   0x77dde091:  mov    0x1c(%ebp),%eax
   0x77dde094:  orl    $0x2,(%eax)
   0x77dde097:  mov    %fs:0x18,%eax
   0x77dde09d:  pushl  -0x8(%ebp)
   0x77dde0a0:  mov    0x30(%eax),%eax
   0x77dde0a3:  push   $0x0
   0x77dde0a5:  pushl  0x18(%eax)
   0x77dde0a8:  call   *0x77dd1394
   0x77dde0ae:  mov    %ebx,%eax
   0x77dde0b0:  pop    %edi
   0x77dde0b1:  pop    %esi
   0x77dde0b2:  pop    %ebx
---Type <return> to continue, or q <return> to quit---q
Quit
(gdb) info thread
  Id   Target Id         Frame
* 1    Thread 1 (CPU#0 [running]) 0x77dddff5 in ?? ()
(gdb)
```

WinDbg


Disassembly - Kernel 'com:pipe,baud=115200,port=\\.\pipe\windbg,resets=...
Offset: @$scopeip
Previous
Next
fc4fd3c0 53 push ebx
fc4fd3c1 56 push esi
fc4fd3c2 8b7508 mov esi,dword ptr [ebp+8]
fc4fd3c5 8b5e28 mov ebx,dword ptr [esi+28h]
fc4fd3c8 57 push edi
fc4fd3c9 8b7d0c mov edi,dword ptr [ebp+0Ch]
fc4fd3cc 68b8110000 push 11B8h
fc4fd3d1 68f4d34ffc push offset CLASSPNP!ClassFindModePage+
fc4fd3d6 57 push edi
fc4fd3d7 56 push esi
fc4fd3d8 e8b3efffff call CLASSPNP!ClassAcquireRemoveLockEx
fc4fd3dd 85c0 test eax,eax
fc4fd3df 57 push edi
fc4fd3e0 56 push esi
fc4fd3e1 0f85e12a0000 jne CLASSPNP!ClassDeviceControlDispatc
fc4fd3e7 8b4360 mov eax,dword ptr [ebx+60h]
fc4fd3ea ff5018 call dword ptr [eax+18h]
fc4fd3ed 5f pop edi
fc4fd3ee 5e pop esi
fc4fd3ef 5b pop ebx
fc4fd3f0 5d pop ebp
fc4fd3f1 c20800 ret 8
fc4fd3f4 643a5c7870 cmp bl,byte ptr fs:[eax+edi*2+70h]
fc4fd3f9 7370 jae CLASSPNP!FreeDictionaryEntry+0x46
fc4fd3fb 5c pop esp
fc4fd3fc 647269 jb CLASSPNP!FreeDictionaryEntry+0x43
fc4fd3ff 7665 jbe CLASSPNP!FreeDictionaryEntry+0x41
fc4fd401 7273 jb CLASSPNP!FreeDictionaryEntry+0x51
fc4fd403 5c pop esp

WinDbg


QEMU
Windows
WinDbg server
WinDbg
QEMU-WinDbg
Windows
WinDbg server
WinDbg

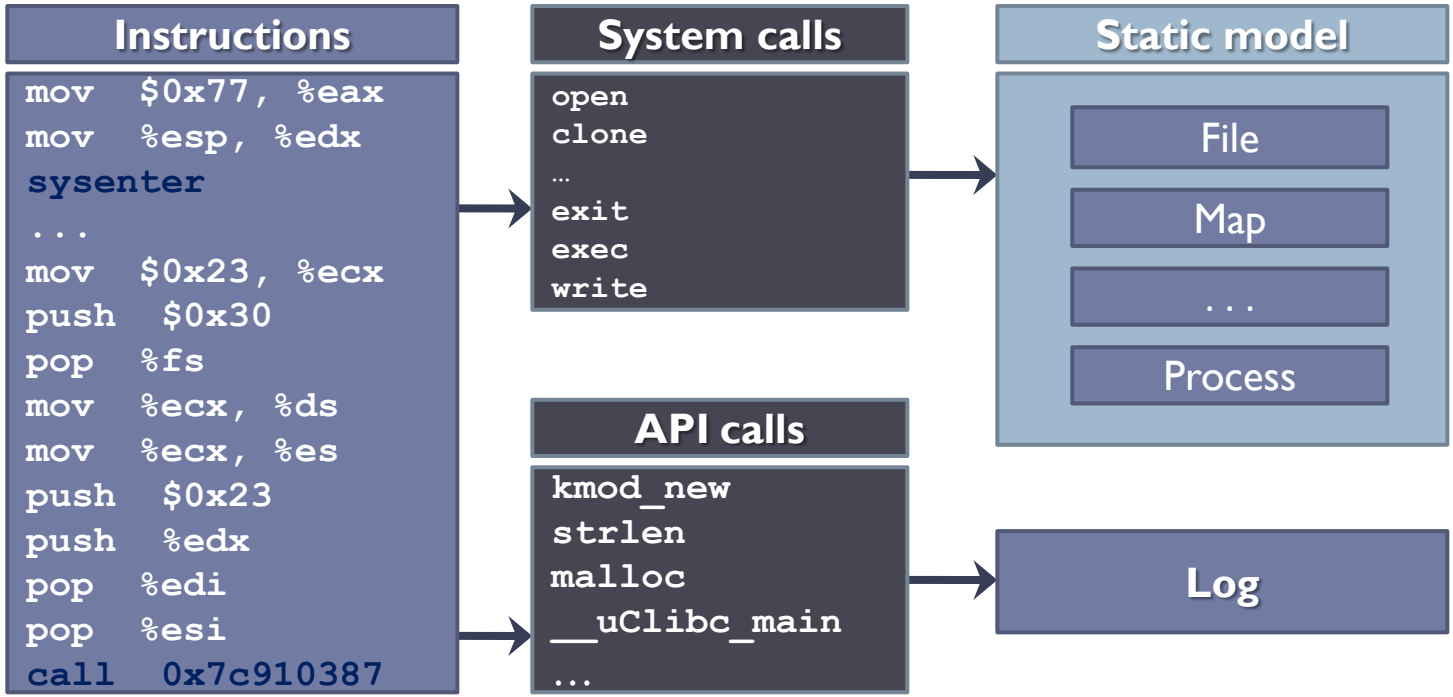
WinDbg

- Отладка не обнаруживается программами
- Работают все обычные команды
- Доступны высокоуровневые возможности WinDbg
- Перехватываются некоторые из исключительных ситуаций

Интроспекция

- Извлечение данных о процессах и потоках
- Трассировка
  - Инструкции
  - Сетевые операции
  - Дисковые операции
- Мониторинг конкретного процесса
  - Системные вызовы
  - Вызовы API-функций

Интроспекция



Интроспекция

- Трассировка машинных инструкций
- Трассировка обращений к диску
- Трассировка отображений виртуальной памяти в физическую
- Отслеживание системных вызовов
- Отслеживание вызовов API-функций
- Сбор информации о процессах
- Извлечение записываемых файлов

Мониторинг ОС на основе Linux/x86

Firmware	Linux kernel version	API tracing overhead, %
AlpineLinux 3.7.0	4.9.65-1	9.2
DD-WRT v24	2.6.23.17	12
EndianFirewall 2.1.2	2.6.9-55	7.1
floppyfw 3.0.14	2.4.37.10	34
IPCop 2.1.8	3.4-3	47
IPFire 2.19	3.14.79	7.5
LEAF 3.1	2.4.34	39
LEAF 6.1.1	4.9.68	6.5
LEDE 17.01.4	4.4.92	1.9
MikroTIK 6.41		39
Openwall 3.1	2.6.18-408	21
OpenWRT 15.05	3.18.23	3.7
Untangle	3.16.0.4	16
ZeroShell 2.0	3.4.6	27

SWAT – System-Wide Analysis Toolkit

- Отладка в gdb и WinDbg
- Запись/воспроизведение работы системы
- Динамическое инструментирование
- Интроспекция

- https://github.com/ispras/swat

Показатели работы записи/воспроизведения

Загрузка ОС	Замедление при записи	Замедление при воспроизведении	Размер журнала, байт на 1000 инструкций
i386 (Win)	31%	156%	2.3
i386 (Debian)	41%	136%	21.9
ARM (Debian)	32%	191%	17.9
MIPS (Debian)	14%	139%	75.4