А.И. Аветисян, А.И. Гетьман, В.А. Падарян, М.А. Соловьев, А.Ю. Тихонов

ВОЗМОЖНОСТИ СРЕДЫ АНАЛИЗА БИНАРНОГО КОДА ТРАЛ И АКТУАЛЬНЫЕ НАПРАВЛЕНИЯ ЕЕ РАЗВИТИЯ.

г. Москва, Институт системного программирования РАН.

ИСП РАН разработана среда анализа защищенного бинарного кода Трал, комбинирующая динамический и статический подходы при анализе защищенного бинарного кода. Возможности среды позволяют автоматизировать решение задач обратной инженерии, в частности:

- восстановление алгоритмов, с получением контрольного примера,
- анализ зловредного кода, в том числе анализ динамики его действий,
- восстановление протоколов в части форматов сообщений и построении модели обработчиков этих сообщений.

Современные средства защиты бинарного кода не позволяют ограничиваться только статическим анализом, исследователь вынужден дополнять свой арсенал как минимум интерактивным отладчиком. Однако больший эффект дает детальная трассировка работы исследуемой программы, поскольку она позволят перейти от интерактивного к post-mortem анализу [1,2], имеющему ряд преимуществ: трасса может многократно использоваться для независимого проведения автоматизированного анализа, результаты трассировки могут анализироваться не одним человеком, а коллективом исследователей, скоростная трассировка позволяет анализировать сетевые программы.

Типичный порядок действий исследователя состоит из следующих шагов. Первым шагом является подготовка стенда, на котором будет получена трасса. Стенд представляет собой виртуальную машину, на которой развернуто необходимое окружение: требуемая операционная система, установлены библиотеки, размещена исследуемая программа. Виртуальная машина запускается в симуляторе, в период работы исследуемой программы происходит трассировка. Применение симулятора позволяет скрывать от исследуемой программы факт трассировки, а соответствующий метод трассировки, благодаря низким накладным расходам, скрывает ее от удаленных машин – возникающее замедление работы не настолько велико, что может быть идентифицировано как отладка/трассировка. Второй шаг – анализ полученной трассы в среде Трал. В качестве промежуточного результата исследователь получает восстановленный по трассе код программы, по которому может быть получен статический образ для последующего анализа в среде IDA Pro.

В настоящее время среда Трал располагает модулями снятия трасс для следующих симуляторов: AMD SimNow™, Dynamips, ARMulator и QEMU. В симуляторе QEMU были проведены существенные доработки, что позволило реализовать в нем т.н. двухпроходную трассировку для архитектуры Intel64. Первый проход создает журнал невоспроизводимых событий, таких как пользовательский ввод, приходящие сетевые пакеты, асинхронные прерывания. Начальное состояние симулируемой системы и журнал затем используются во втором проходе, представляющем собой детерминированное воспроизведение [3] работы системы, на этом проходе собирается детальная трасса. Преимущества такого метода трассировки в крайне малых накладных расходах при выполнении первого этапа, что позволяет получать трассы сетевых программ, оснащенных развитыми средствами защиты от анализа.

Текущие работы по развитию среды были сформированы, исходя из следующих перспективных направлений.

В настоящее время анализу повергается одна трасса. Исследователь, анализируя программу, выделяет (как правило, автоматически) подтрассы. Цель такого выделения – сократить количество анализируемых инструкций. Однако исходная трасса, содержащая полный набор выполнявшихся компьютером инструкций, только одна. Предполагается дать возможность в рамках одного исследовательского проекта согласованно совмещать произвольное количество трасс. Это позволит улучшать качество покрытия кода, отслеживать взаимодействие кода работающего на разных процессорах (в том числе разной архитектуры), что весьма актуально для коммуникационного оборудования. Такое расширение возможностей ставит задачу доработки сопряжения с IDA Pro, поскольку эта среда может работать только с образом одной программы.

Одним из преимуществ интерактивного отладчика по сравнению с трассой является возможность доступа к произвольному месту памяти исследуемой программы или системы, если используется отладчик уровня ядра. Предполагается интегрировать в среду Трал симулятор, обладающий интерфейсом интерактивного отладчика. Симулятор использует бинарную трансляцию кода целевой архитектуры с применением промежуточного представления Pivot [4], для которого уже поддерживается трансляция кода нескольких целевых архитектур: Intel64, MIPS64, ARM v6, PowerPC32. Помимо того, интерпретация незадействованных ветвей улучшит покрытие кода трассами.

Организация файлов исследовательского проекта в виде БД с единым интерфейсом позволяет обойти еще одно узкое место: распространение знаний об анализируемой программе между несколькими исследователями. Появляется возможность одновременно задействовать коллектив исследователей, которые смогут распределить участки кода между собой, а затем работать, синхронизируя полученные данные с другими исследователями. К числу распространяемых данных относятся как результаты автоматизированного анализа, полученные при запусках соответствующих алгоритмов на своем компьютере, так и свое формально представленное понимание работы исследуемого кода.

Литература

1. Андрей Тихонов, Арутюн Аветисян, Вартан Падарян. Методика извлечения алгоритма из бинарного кода на основе динамического анализа. // Проблемы информационной безопасности. Компьютерные системы. №3, 2008. стр. 66-71.
2. В.А. Падарян, А.И. Гетьман, М.А. Соловьев. Программная среда для динамического анализа бинарного кода. // Труды Института Системного Программирования. Том 16. 2009. Стр. 51-72.
3. M. Xu, V. Malyugin, J. Sheldon et al. ReTrace: Collecting execution trace with virtual machine deterministic replay / VMware Inc.— 2008. http://govirtual.org/docs/DOC-1321.
4. В.А. Падарян, М.А. Соловьев, А.И. Кононов. Моделирование операционной семантики машинных инструкций. // Труды Института системного программирования РАН, Том 19, 2010. Стр. 165-186.