ИСП CRUSHER:
КОМПЛЕКС ДИНАМИЧЕСКОГО АНАЛИЗА ПРОГРАММ

ИСП Crusher — программный комплекс, комбинирующий несколько методов динамического анализа. Состоит из двух инструментов: ИСП Fuzzer для проведения фаззинг-тестирования и SyDr, отвечающий за автоматическую генерацию тестов для сложных программных систем. В ближайшие 1-2 года в комплекс планируется включить ещё две технологии ИСП РАН: BinSide и Casr. ИСП Crusher позволяет построить процесс разработки всоответствии с ГОСТ Р 56939-2016 и «Методикой выявления уязвимостей и недекларированных возможностей в программном обеспечении» ФСТЭК России.

ИСП РАН

ИНСТРУМЕНТ ТЕСТИРОВАНИЯ ИСП FUZZER

ИСП Fuzzer — инструмент проведения фаззинг-тестирования, необходимый на всех этапах разработки, тестирования и эксплуатации ПО. Обнаруживает ошибки или закладки как при наличии, так и при отсутствии исходного кода. Решает те же задачи, что и мировые аналоги (Synopsys Codenomicon, beSTORM, Peach Fuzzer), однако более удобен для российских компаний в условиях процесса импортозамещения.

ИСП FUZZER — ЭТО:

- Осуществление фаззинг-тестирования через различные источники внешних данных (файл, аргументы командной строки, стандартный поток ввода, аргументы переменных окружений, сеть).
- Возможность добавления пользовательских мутационных преобразований (для генерации новых входных данных и увеличения эффективности тестирования).
- Наличие модулей пред- и постобработки входных данных для осуществления константных преобразований над данными перед их отправкой в анализируемое ПО.
- Поддержка многопоточного анализа как на одной машине, так и на распределённых.
- Поддержка пользовательских плагинов отправки данных по сети (плагины позволяют осуществлять взаимодействие с клиентским или серверным ПО и отправлять мутированные данные).
- Возможность интеграции с рядом необходимых инструментов жизненного цикла разработки безопасного ПО, созданных в ИСП РАН:
  - использование инструментов динамического символьного выполнения SyDr и Anxiety для повышения эффективности фаззинг-тестирования;
  - возможность получать входные данные, на которых проявляются ошибки, размеченные инструментом статического анализа BinSide в автоматическом режиме;
  - отображение трассы последовательности функций, приводящих к аварийному завершению, в интерфейсе статического анализатора Svace;
  - использование генератора данных по формальным грамматикам ANTLR для формирования корпуса входных данных;
- Совместная работа с дизассемблером IDA PRO (сохранение покрытия для плагина Lighthouse, которое отображает покрытые базовые блоки в ПО, а также вывод процента покрытых базовых блоков).

— Возможность проведения анализа серверного и клиентского ПО, работающего по протоколам с состояниями и без состояний.
— Возможность запуска систем динамического анализа на новых входных данных (Valgrind, DrMemory, QASan).
— Лёгкая расширяемость и добавление новых методов в рамках существующей инфраструктуры; оперативная адаптация под новые задачи.
— Возможность распределения корпуса входных данных между процессами фаззеров для повышения эффективности их работы.
— Оценка критичности найденных аварийных завершений.
— Использование фаззера Radamsa для генерации новых данных.
— Поддержка различных видов инструментации: DynamoRIO, QemuUserMode, статическая инструментация GCC, статическая инструментация LLVM, QemuSystemArm.

ИНСТРУМЕНТ ДИНАМИЧЕСКОГО СИМВОЛЬНОГО ВЫПОЛНЕНИЯ SYDR

SyDr — инструмент автоматической генерации тестов для сложных программных систем с целью увеличения покрытия кода и обнаружения ошибок. Строит математическую модель программы, позволяя фаззеру открывать новые пути выполнения, которые сложно обнаружить классическими методами генетических мутаций.Разработанные методы развивают технологию символьного выполнения, представленную в созданных ранее в ИСП РАН инструментах Avalanche и Anxiety. В отличие от аналогичных открытых инструментов, SyDr проверяет результаты своей работы на корректность и определяет, действительно ли сгенерированные входные данные приводят к инвертированию целевых переходов.

SYDR — ЭТО:

условных переходов, которые зависят от входных данных. Реализована возможность параллельного инвертирования.
— Интеграция с ИСП Fuzzer для инвертирования, что решает проблемы при прохождении переходов, зависящих от сравнения с константами.
— Решение задач обратной разработки. Помощь аналитику в достижении интересующей его точки в программе. Получение трассы инструкций, которые зависят от входных данных.
— Поддержка различных источников внешних данных программы (файлы, сетевые сокеты, переменные окружения, стандартный поток ввода, аргументы командной строки).
— Предикаты безопасности. Генерация входных данных, приводящих к проявлению дефекта (деление на ноль, разыменование нулевого указателя, выход за границы буфера).
— Символьное выполнение многопоточных программ.
— Инвертирование косвенных переходов (switch statement). Разработан алгоритм определения таблиц и переходов по вычисляемым адресам.
— Слайсинг формул. Удаление избыточных формул из предиката пути, которые не влияют на инвертируемый условный переход. Решает проблему недостаточной помеченности, а также ускоряет обработку запросов SMT-решателем.

ДЛЯ КОГО ПРЕДНАЗНАЧЕН ИСП CRUSHER?

— Компании, нацеленные на разработку ПО с высоким уровнем надёжности и безопасности.
— Компании, отвечающие за аудит или сертификацию ПО.

СИСТЕМНЫЕ ТРЕБОВАНИЯ

Поддержка ОС семейства Linux и Windows. Возможность проводить фаззинг-тестирование встроенных устройств (контроллеры, устройства интернета вещей), а также сервисов и COM-объектов ОС Windows.

ОПЫТ ВНЕДРЕНИЯ

ИСП Crusher в разной комплектации используется более чем в 20 компаниях и лабораториях, в том числе в ОАО «РусБИТех», Postgres Professional, ООО «Код Безопасности», МВП «СВЕМЕЛ» и др.

СХЕМА РАБОТЫ ИСП CRUSHER