Исследование методов анализа программного обеспечения без использования исходного кода

Велижанин Анатолий Сергеевич
Тюменский государственный нефтегазовый университет (Тюмень), Россия
e-mail: Anatoliy.Velizhanin@gmail.com

Ревнивых Александр Владимирович
Тюменский государственный нефтегазовый университет (Тюмень), Россия

Аннотация

Методы исследования программного обеспечения без использования исходных кодов

Современное программное обеспечение состоит из множества программных модулей, реализованных с применением различных технологий. Рассматривая тенденции развития технологической базы разработки программного обеспечения отметим движение в сторону платформы .NET Framework для Microsoft Windows. В основе функционирования программных решений, разработанных под платформу .NET Framework лежит байт-код. Принцип JIT компиляции в данной технологии несколько схож с подходом, реализованным для языка программирования Java. Для UNIX-подобных операционных систем преимущественно разрабатывается “Native” (скомпилированное под платформу) программное обеспечение. Отдельно следует обратить внимание на Android системы в основе которых лежит ядро Linux. Значительная часть программных приложений для этой операционной системы разрабатывается для работы под Dalvik виртуальной машиной.

Многие программные решения распространяются без исходных кодов, к тому же настройки компиляторов в значительной степени влияют на результирующий исполняемый файл. Более того, некоторые программные модули разработаны для выполнения под управлением какой-либо виртуальной машины и скомпилированы в байт-код. Тогда дополнительное влияние на результирующие выполняемые машинные инструкции оказывает и версия виртуальной машины. Таким образом возрастает важность анализа программного обоеспечения без использования исходных кодов.

В настоящее время ряд средств позволяет проводить реверс-инжиниринг как Native программного обеспечения, так и базирующегося на байт-коде. К таким средствам относятся Hex Rays IDA Pro, Red Gate .NET Reflector, JetBrains dotPeek и другие.

Важно отметить, что вышеприведенные средства применяются для анализа исполняемых файлов на диске. Однако, учитывая современное разнообразие технологий разработки программного обеспечения, загруженное и инициализировавшее внутренние структуры данных приложение может несколько отличаться от образа на диске. К тому же, совмещение модулей, реализованных в форме байт-кода с Native компонентами программного решения, может привести к значительному усложнению процесса качественного анализа. Причиной тому является, в том числе, JIT-компиляция. Виртуальные машины различных версий могут генерировать несколько различный машинный код. Та-

ким образом возрастает важность анализа фактически сформированного в ходе JIT-компиляции машинного кода, который зачастую совмещен с программным кодом Native модулей.

В качестве одного из вариантов решения данной задачи является анализ дампов памяти, снятых с приостановленного в интересующий момент времени процесса.

Работа посвящена исследованию методов анализа программного обеспечения без использования исходных кодов с учетом современных тенденций развития технологий разработки программных решений.