0.1. *Лебедев Р.К., Корякин И.А.* Метод защиты программ при помощи переключения режимов исполнения архитектуры x86

Одним из основных методов обратной разработки программного обеспечения является дизассемблирование, являющееся также одним из шагов в других видах анализа: например, символьном исполнении и декомпиляции. В связи с этим, задача противодействия ему является актуальной, так как может помочь защитить программы от обратной разработки и ее нежелательных последствий.

Для многих процессорных архитектур дизассемблирование — достаточно простой и однозначный процесс, что делает задачу противодействия ему без использования самомодифицирующегося кода практически неразрешимой. Однако для архитектуры x86, широко используемой в персональных компьютерах и серверах, это не так: машинные коды x86 позволяют записывать идентичные инструкции разными способами, некоторые машинные коды вовсе не имеют представления в виде инструкций [1], а также код может динамически переключаться между 32-битным и 64-битным режимами исполнения в одной программе [2].

В данной работе предложен метод защиты, использующий динамическое переключение между 32-битным и 64-битным режимами и несоответствия между машинными кодами в данных режимах. Эксплуатируется неспособность дизассемблера в рамках статического анализа надежно определить, в каком режиме исполняется конкретный участок кода, что может привести к ложному восприятию кода дизассемблером и использующими его вывод инструментами.

Предложенный метод основан на добавлении в программу конструкций, являющихся корректным машинным кодом как в 32-битном, так и в 64-битном режиме, но исполняющихся по-разному в зависимости от режима исполнения. Хотя большинство машинных кодов не потеряли своего значения с переходом к 64-битной версии архитектуры x86 (с точностью до размера регистра), есть и исключения, например пара из инструкции INC EAX (32-битный режим) и префикса REX (64-битный режим). Они кодируются одним шестнадцатеричным байтом 40, однако имеют совершенно разный смысл: INC EAX увеличивает регистр на единицу, а REX влияет на размер используемого следующей инструкцией регистра, причем если инструкция не использует регистры, он просто игнорируется. Соответственно, если разместить после данной инструкции условный переход, зависящий от значения флага ZF, изменяемого (или нет) соответствующей инструкцией или префиксом, осуществление перехода будет зависеть от режима исполнения, что при несовпадении ожиданий дизассемблера с реальностью отправит его по ложному следу.

Метод был реализован на уровне ассемблерных преобразований и показал свою эффективность против декомпиляторов Ghidra и IDA, а также инструмента символьного исполнения angr, вызвав ложную декомпиляцию и исполнение ими кода программы соответственно. Значимой особенностью метода является отсутствие самомодифицирующегося кода, что делает его применимым для операционных систем с самыми строгими политиками безопасности, не позволяющих динамическую модификацию исполняемого кода.

Научный руководитель — д.т.н. Павский К. В.

Список литературы

[1] XlogicX Assembly Language is Too High Level // DEF CON 25, 2017.

[2] The 0x33 Segment Selector (Heavens Gate). [Электронный ресурс]. URL: https://www.malwaretech.com/2014/02/the-0x33-segment-selector-heavens-gate.html (дата обращения 01.09.2022).