Идентификация статически слинкованных библиотек в исполняемых файлах

Арутюнян Мариам Сероповна

Аспирант

Институт математики и информатики РАУ, Ереван, Армения

E-mail: arutunian@ispras.ru

Научный руководитель *— Аветисян Арутюн Ишханович*

В среднем, библиотечные функции составляют от пятидесяти до девяноста процентов программы. Использование сторонних библиотек при написании программы облегчает работу программиста, но они также могут создавать серьезные угрозы безопасности и препятствовать анализу программ. Также в старой версии библиотеки может быть ошибка, которая исправлена в новой версии, но при этом в программе используется старая версия. Поэтому в программе нужно будет использовать новую версию библиотеки.

Часто потребитель имеет только исполняемый код программы, и неизвестно, что там написано. Исполняемые файлы часто статически связаны с библиотеками. Уязвимость в одной из этих библиотек может привести к эксплуатации устройства, особенно если учесть, что эти библиотеки часто обрабатывают данные, контролируемые пользователем. Таким образом, определив библиотеки и их версии в исполняемом коде, можно решить вопрос использования безопасных библиотек.

Для определения, какие библиотеки слинкованы в исполняемом файле, инструмент пытается сопоставить функции библиотек с функциями исполняемого файла. Если функции конкретных библиотек полностью сопоставляются с некоторыми функциями исполняемого файла, предполагается, что эта библиотека статически слинкована.

Для идентификации библиотек собрана база из разных версий часто используемых библиотек. Каждая библиотека из собранной базы сравнивается с исполняемым файлом. Сначала библиотеки и исполняемый файл дизассемблируются и транслируются в промежуточное представление REIL [2]. Потом происходит генерация графа зависимостей программы [3] и графа вызовов функций. Далее на основе этих графов инструмент пытается сопоставить функции каждой библиотеки и исполняемого файла. Для сопоставления функций используется метод, представленный в статье [1]. В итоге инструмент выдает те библиотеки и их версии, которые предположительно

статически слинкованы в исполняемом файле.

Литература

1. Aslanyan H., Avetisyan A., Arutunian M., Keropyan G., Kurmagaleev Sh., Vardanyan V. Scalable Framework for Accurate Binary Code Comparison // Ivannikov ISPRAS Open Conference (ISPRAS). Moscow, Russia, 2017, P. 34–38.
2. Dullien T., Porst S. REIL: A platform-independent intermediate representation of disassembled code for static code analysis // In Proceedings of the CanSecWest Conference. 2009.
3. Ferranite J., Karl J., Joe D. The Program Dependence Graph and Its Use in Optimization // ACM Transactions on Programming Languages and Systems. 1987. V. 9, № 3, P. 319–349.