Н. П. Цыганенко
(БГТУ, Минск)

СТАТИЧЕСКИЙ АНАЛИЗ КОДА ДЛЯ ВЫЯВЛЕНИЯ УЯЗВИМОСТЕЙ С ИСПОЛЬЗОВАНИЕМ ROSLYN

Статический анализ кода – это проверка исходного кода приложения без реального выполнения на соответствие определенному набору правил. Такой вид анализа может использоваться для выявления ошибок и уязвимостей в исходном коде программного обеспечения. Программа выполняющая статический анализ называется статическим анализатором.

Анализ может проводиться либо над исходным кодом либо над объектным (MSIL, байт-код). Статический анализ не гарантирует 100%

нахождения ошибок и уязвимостей. Возможны ложные (false positive) и ложноотрицательные срабатывания (false negative). По этой же причине статический анализатор в общем случае не предназначен для исправления найденных ошибок и уязвимостей, он предупреждает программиста о подозрительных и потенциально проблемных участках кода [1].

Статический анализатор можно отнести к прикладному классу средств защиты. С помощью моделирования уязвимостей с последующим их внедрением в приложения, а также разработки алгоритма их определения можно построить статический анализатор, который будет выявлять уязвимости в приложениях для мобильных систем. Такой анализатор может использовать механизм синтаксических деревьев с целью обеспечения более высокоуровневой работы с исходным кодом при его анализе на соответствие условиям алгоритма выявления определённой уязвимости [2].

Синтаксическое дерево – это конечное, помеченное, ориентированное дерево, в котором внутренние вершины сопоставлены с операторами языка программирования, а листья – с соответствующими операндами.

Если речь идёт о статическом анализаторе кода написанном на языке программирования C#, то для построения синтаксических деревьев отлично подходит компилятор с открытым исходным кодом Roslyn от Microsoft. Он позволяет проверять условия алгоритма выявления уязвимости в удобной для программиста форме, с использованием возможностей технологии LINQ.

ЛИТЕРАТУРА

1. Chess, B. Secure Programming with Static Analysis / B. Chess – Boston: Addison-Wesley Professional, 2007. – 624 p.

2. Boulanger, J. Static Analysis of Software: The Abstract Interpretation / J. Boulanger – Indianapolis: Wiley, 2011. – 331 p.