СЕКЦИЯ 3. СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ И ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЯХ

ОБФУСКАЦИЯ JAVASCRIPT КОДА

М.А. БАРТОШИК

Обратное проектирование позволяет анализировать и модифицировать механизмы защиты приложений с целью их несанкционированного использования, что указывает на необходимость разработки технических методов защиты исходного кода.

Лексическая обфускация включает в себя изменение имен идентификаторов, строковых констант, форматирования кода и использование нестандартных кодировок. Данный метод усложняет процесс обратного проектирования, уменьшает объем кода и может повысить производительность приложения.

Методы синтаксической обфускации основываются на модификации потока управления программы. Данные методы, как правило, приводят к ухудшению производительности и не могут применяться в критических для скорости выполнения участках кода. Среди наиболее эффективных методов можно выделить: применение непрозрачных предикатов, добавление «мертвого» и избыточного кода, смешивание и клонирование функций, вставка операторов функций по месту вызова и вынос группы операторов в отдельную функцию, изменение области действия переменных, развертка циклов, подмена вызовов встроенных функций, использование нестандартных способов исполнения кода (eval, Function, setTimeout).

При применении обфускации необходимо учитывать особенности JavaScript: динамическая типизация, доступность исходного кода злоумышленнику, возможность автоматической модификации кода через прокси-сервер, наличие инструментов отладки и анализа потока управления, отличия реализации стандартных функций в браузерах.

В общем случае применение обфускации дает лишь временную защиту приложения от взлома. В случае с JavaScript существует возможность применять различный случайный набор преобразований с определенной периодичностью (например, раз в сутки). Такой подход может сделать взлом приложения нерентабельным.

К РЕШЕНИЮ УРАВНЕНИЙ НАД ПОЛЯМИ ГАЛУА. СОГЛАСОВАНИЕ ПРИМИТИВНЫХ ЭЛЕМЕНТОВ КОНЕЧНОГО ПОЛЯ И ЕГО РАСШИРЕНИЯ

В.А. БОГРЕЦОВ, В.А. ЛИПНИЦКИЙ

Теория конечных полей Галуа играет большую роль в современной теории помехоустойчивого кодирования и криптографии. Особое место занимают в данных областях алгоритмы решения полиномиальных уравнений над названными полями. Общеизвестно, например, что в процессе декодирования БЧХ-кода, наиболее ресурсоёмкой процедурой является нахождение корней полинома локаторов ошибок [1]. Ещё одним интересным примером, показывающим значимость эффективных алгоритмов решения полиномиальных уравнений в конечных полях, является криптосистема McEliece [2], основанная на применении двоичных кодов Гоппы. Одним из этапов расшифровки сообщения в данной системе является решение полиномиального уравнения над полем Галуа. Данный этап занимает до ¾ всего времени процесса расшифровки. В [3] описан алгоритм решения кубического уравнения в поле Галуа характеристики 2. В основе данного алгоритма лежит сведение исходного уравнения к квадратному, которое может быть решено одним из методов, описанных в [4]. Затем, по найденным корням квадратного уравнения, восстанавливаются корни исходного уравнения. Если степень исходного поля