ИНСТРУМЕНТЫ АНАЛИЗА РИСКОВ В JAVASCRIPT

И.Д. Стаселько, М.А. Аниховский, Ю.И. Алексеев, А.С. Летохо

В наши дни практически невозможно писать на JavaScript без использования одной из тысяч доступных библиотек JavaScript с открытым исходным кодом. Библиотеки делают кодирование в JavaScript эффективнее, упрощая процессы, которые требуют нескольких строк кода для их достижения. Однако эти преимущества не обходятся без рисков.

Библиотеки JavaScript уязвимыми. Согласно исследованию, Северо-Восточного университета, «более 37 % веб-сайтов используют хотя бы одну версию библиотеки с известной уязвимостью» [1]. Уязвимости безопасности в JavaScript включают межсайтовый скриптинг (возможность внедрить вредоносный код) «Экосистема JavaScript не имеет надежной структуры для документирования уязвимостей в библиотеках и документирования их последствий», – сказал SD Times Арнал Дайаратна, директор по исследованиям IDC.

Нет единого списка уязвимостей, доступных для разработчиков. Уязвимости JQuery отображаются на веб-сайте CVE (каталогом распространенных уязвимостей). Тем не менее, Angular не отображается в CVE; вместо этого он использует GitHub CHANGELOG для сообщения об уязвимостях безопасности, каждая библиотека обрабатывает информацию о безопасности по-разному, поэтому разработчики не могут полагаться на одно местоположение обновлений списка угроз. Согласно сообщению, в блоге npm: «Современный проект JavaScript обычно зависит от 700–1200 пакетов». Именно здесь инструменты анализа состава программного обеспечения становятся необходимыми. Инструменты анализа проверяют код и выбирают уязвимые компоненты, например, инструмент «SonarQube». Это ускоряет процесс обнаружения уязвимостей на сайте, а также снижает риск человеческих ошибок. Исследование Северо-Восточного университета показало, что медианный веб-сайт использовал библиотечную версию, которая была «на 1177 дней старше, чем новейшая версия».

Переход на более новую версию, учитывающую потенциальную угрозу, требует времени, поскольку необходимо выполнить тестирование, чтобы убедиться, что последняя версия совместима с существующим приложением или сайтом. Постоянное обновление до более новых версий библиотек не может гарантировать полную защиту существующим сайтам от уязвимостей, что отменит любую экономию средств, полученную от неправильного обновления сайта.

Литература

1. K. Watanabe, T. Fukamachi, N. Ubayashi and Y. Kamei, PosterIEEE International Conference on Software Testing [Electronic resource]. – Access mode: https://aws.amazon.com/solutions/casestudies. – Date of access: 29.11.2019.