

УДК 004.9

## **РАЗРАБОТКА ПРИЛОЖЕНИЯ ВОССТАНОВЛЕНИЯ ПАРОЛЯ ДЛЯ БРАУЗЕРА OPERA**

Артемов Г.И., студент группы ИСт-212, II курс

Назимов А.С. к.т.н., доцент каф. ИиАПС

ФГБОУ ВО «Кузбасский государственный технический университет  
имени Т.Ф. Горбачева»  
г. Кемерово

Как правило, у пользователя персонального компьютера имеется в наличии большой набор различных программ, и зачастую самая востребованная из них - это браузер. Браузер является прикладным программным обеспечением, которое позволяет просматривать интернет-ресурсы.

Браузеры используют для взаимодействия с веб-сайтами в глобальной сети Интернет. Современные браузеры поддерживают протокол FTP и могут использоваться непосредственно для просмотра файлов различных типов: графических, тестовых, аудио и видео файлов. Возможности браузеров постоянно расширяются благодаря их постоянному совершенствованию, популярности IT и конкуренции между разработчиками программного обеспечения.

Расширения появились благодаря браузеру Mozilla Firefox [1], который был разработан как небольшая программа для веб-серфинга, а его функциональные возможности могли быть расширены посредством дополнительных модулей (расширений). Таким образом, расширение - это небольшая компьютерная программа, которая позволяет расширить стандартные возможности браузера. Термин «расширение», в зависимости от браузера, может трактоваться по-разному. Он может называться плагином (plug-in), аддоном (add-on) или расширением (extension). Следует отметить, что расширения и аддоны это не одно и то же. Аддон - это, например, Flash-анимация, которая с 2020 года находится в фазе удаления из браузеров, так как имеет в себе уязвимости. Для разработки расширений используются HTML, CSS и JavaScript, а также API браузера, чтобы, например, внедрять код расширения в страницу пользователя. С 2019г. стало возможным использовать один и тот же код в разных браузерах (Opera, Google Chrome и так далее). Из этого следует, что расширения из одного браузера можно использовать в любом другом браузере.

Существует невероятно огромное количество расширений для различных задач. Например, блокировщик рекламы Adblock для Chrome [2], загрузчик видео с видео-хостинга YouTube для MozillaFirefox [3] и другие. Также особо следует отметить расширения, которые блокируют браузер в целях безопасности (например, от действий незнакомых людей и детей). Таких расширений существует немало, но хотелось бы выделить расширение для браузера

Opera, которое называется «Setpasswordforyourbrowser (Operalock )» [4, 5].

При установке этого расширения появится форма для установки пароля, после чего браузер закроется. При открытии браузера будет запрашиваться пароль, если попытаться закрыть форму ввода пароля или ввести неправильный пароль - браузер закроется. Данное решение предоставляет защиту от несанкционированного доступа к браузеру пользователем сторонним лицом.

В случае потери пароля от этого расширения пользователем возникает необходимость его восстановления или замены. Для этого требуется специализированное программное решение, которое восстановит и/или заменит пароль, так как в большинстве расширений не предусмотрено восстановление пароля.

Для разработки собственного программного решения по восстановлению или замены пароля (на примере расширения «Setpasswordforyourbrowser (Operalock)») использовались следующие технологии: C# и Node.js [6, 7]. На языке программирования C# реализовано взаимодействие с пользователем (интерфейс программы) и со скриптами Node.js, которые в свою очередь взаимодействуют с хранилищем данных расширения, для восстановления или замены его пароля. Аналогичным образом можно изменить иные настройки других расширений (например, в расширении LockPW можно изменить поле с количеством попыток ввода пароля [8]), меняя соответствующие поля в хранилище данных расширения. Для управления настройками различных расширений необходим индивидуальный подход, так как у каждого расширения уникальные наименования полей и тип их шифрования. Например, пароль может храниться в «чистом» незашифрованном виде, а может быть зашифрован каким-нибудь собственным алгоритмом. В этом случае для этого необходимо проанализировать исходный код расширения, чтобы переписать/портировать функцию шифрования в программу.

Как правило, хранилище данных расширения имеет простую структуру - это простая база данных, которая состоит из кортежей «ключ-значение». Обращение в базу данных производится через специальную библиотеку Node.js.

Рассматривая структуру любого расширения, можно выделить файл, который должен быть включен в обязательном порядке, а именно - файл манифеста (manifest.json). Он является «точкой входа» в расширение. По спецификации файл манифеста является валидным JSON-файлом. Полное описание ключей, которые могут быть использованы в манифесте в зависимости от используемого браузера, можно узнать в спецификации. Использование ключей, которых нет в спецификации, приведет к их игнорированию (например, в браузерах Firefox и Chrome появляются ошибки, но расширение продолжает свою работу) [9].

Расширение имеет три контекста исполнения кода, то есть можно сказать, что приложение состоит из трех частей (Extension context, Content script context и Web page context) с разным уровнем доступа к браузерному API. В Extension context доступна большая часть API. Также в этом контексте содер-

жаться: background page («backend» часть расширения), popup page (страница, которая появляется при нажатии на иконку расширения), custom page (страница расширения, существующая в специальной отдельной вкладке вида «chrome-extension://<id\_расширения>/customPage.html»). Рассматривая уровень Content script context, можно понять, что файл контент-скрипта запускается вместе с каждой вкладкой браузера. Именно контент-скрипты взаимодействуют с содержанием веб-страницы. Уровень Web page context - сама веб-страница. К расширению она не имеет никакого отношения и доступа туда не имеет, кроме случаев, когда в манифесте явно не указан домен этой страницы. Также у расширений существует API для взаимодействия между разными частями приложения [9, 10].

Разработанное расширение по восстановлению и/или замене пароля создавалось в первую очередь в учебных целях для освоения возможностей прикладного программирования расширений для браузеров с использованием технологий C# и Node.js .

В заключение хочется сказать, что расширения для блокировки браузера не являются достаточной защитой для браузера. Наиболее эффективными способами являются использование встроенной функции мастер-пароля или комплексный способ защиты браузера.

### Список литературы:

1. Браузер Mozilla [Электронный ресурс] // Режим доступа: <https://www.mozilla.org/> (Дата обращения 21.02.2023).
2. AdBlock - лучший блокировщик рекламы [Электронный ресурс] // Режим доступа: <https://chrome.google.com/webstore/detail/adblock-best-ad-blocker/gighmmpiobklfepjocnamgkkbiglidom?hl=ru> (Дата обращения 21.02.2023).
3. SaveFrom.net помощник: скачать Ютуб, ВКонтакте и др. [Электронный ресурс] // Режим доступа: <https://addons.mozilla.org/ru/firefox/addon/savefromnet-helper> (Дата обращения 10.02.2023).
4. Браузер Opera [Электронный ресурс] // Режим доступа: <https://www.opera.com/ru> (Дата обращения 21.02.2023).
5. Set password for your browser ( Opera lock ) [Электронный ресурс] // Режим доступа: <https://addons.opera.com/ru/extensions/details/set-password-for-your-browser-opera-lock> (Дата обращения 21.02.2023).
6. C#[Электронный ресурс] // Режим доступа: <https://dotnet.microsoft.com/en-us/languages/csharp> (Дата обращения 21.02.2023).

7. Node.js [Электронный ресурс] // Режим доступа: <https://nodejs.org/ru/>  
(Дата обращения 21.02.2023).

8. LockPW [Электронный ресурс] // Режим доступа:  
<https://chrome.google.com/webstore/detail/lockpw/kjcloihghgncbdkaafgkckbokjnehfmo> (Дата обращения 21.02.2023).

9. Структура расширения [Электронный ресурс] // Режим доступа  
<https://habr.com/ru/company/waves/blog/451796/> (Дата обращения 21.02.2023).

10. Введение в web APIs [Электронный ресурс] // Режим доступа  
[https://developer.mozilla.org/ru/docs/Learn/JavaScript/Client-side\\_web\\_APIs/Introduction](https://developer.mozilla.org/ru/docs/Learn/JavaScript/Client-side_web_APIs/Introduction) (Дата обращения 21.02.2023).