А.А. Мурзакова, Е.А. Мурзакова, О.В. Желудкова, М.М. Акимова
Тюменский государственный университет, г. Тюмень

УДК 004.056

БЕЗОПАСНОСТЬ PWA-ПРИЛОЖЕНИЙ НА ПРИМЕРЕ РЕАЛИЗАЦИИ СЕРВИСА S2W

Аннотация. В статье рассмотрены преимущества и аспекты безопасности новой технологии – гибрида мобильного приложения и сайта – Progressive Web Application на примере реализации сервиса S2W.

Ключевые слова: гибрид мобильного приложения и сайта, service worker, push-оповещения, offline режим, https протокол

Целью данной работы было изучить технологию Progressive Web Apps и показать безопасные стороны на примере разработки собственного прототипа PWA-приложения для сервиса S2W.

Чтобы понять, что такое Progressive Web Apps (PWA), достаточно представить, что некий сайт взаимодействует с пользователем как приложение. То есть у пользователя есть возможность установить его на любой гаджет, получать уведомления и работать с ним. Причем работа может продолжаться даже без Интернет-соединения.

Эта новая модель приложения пытается объединить функции, предлагаемые большинством современных браузеров, с преимуществами мобильной работы.

PWA пришел следом за такой относительно недавней (осень 2015 г.) разработкой Google, как AMP (Accelerated Mobile Pages). AMP (Accelerated Mobile Pages) – разработка Google, основными задачами которой выступают оптимизация и ускорение загрузки контентных страниц (статей, новостей, обзоров и т.д.). Основным преимуществом AMP выступает скорость, а PWA – высокая динамичность и расширенные возможности платформ. [1]

Согласно данным Google Developers характеристиками PWA должны выступать: безопасность (PWA подается через HTTPS для предотвращения слежки и подделки содержания), прогрессивность (PWA работает для каждого пользователя, независимо от выбора браузера), эффективность (PWA поддерживают настольные системы, мобильные устройства, планшетные, и др.), независимое подключение (PWA имеют возможность работать в автономном режиме или в сети низкого качества), подобность приложениям (пользователь воспринимает PWA как обычное приложение с привычным взаимодействием и навигацией, за счет того, что PWA основано на модели оболочки обычного приложения), модернизированность (постоянная обновляемость, благодаря обновлениям Service worker), обнаруживаемость (PWA идентифицируется как «приложение», благодаря W3C манифестам и Service worker, что позволяют поисковым системам их находить), повторное соединение (организуется через push-уведомления), процесс установки (пользователи могут хранить наиболее нужные приложения на домашнем экране без взаимодействия с app store, к примеру), связуемость (обмен производится через URL-адрес. К тому же не требуется сложная установка). [2]

PWA – это гибрид, сочетающий лучшие стороны web-сайтов и мобильных приложений.

Ниже рассмотрены отличия PWA от веб-сайтов и мобильных приложений.

Отличия PWA от WEB-сайтов

В своем ядре PWAs ничем не отличаются от обычных веб-сайтов – они также сделаны из HTML, CSS и JavaScript и находятся в браузере. [3]

Отличают PWA от обычных веб-сайтов такие характеристики, как безопасность, прогрессивность, эффективность и т.д. (10 основных характеристик PWA-приложений).

Отличия PWA от native мобильных приложений

1. В браузерах Google, Opera, Firefox и Microsoft PWA можно загрузить на любых гаджетах, независимо от размера экрана и других спецификаций. Кроме того, разработчики данных браузеров будут предлагать пользователям установить PWA при втором посещении сайта.

2. Разработчики приложений отмечают, что создать прогрессивное приложение легче и быстрее, чем обычный сайт.

3. Не нужно API с поддержкой обратной совместимости. В случае с PWA пользователи запускают ту же версию кода сайта (в отличие от классических приложений).

Актуальность данной технологии заключается еще в устранении проблемы резкого снижения спроса людей на установку приложений. По исследованиям, большинство пользователей смартфонов скачивают ноль приложений в месяц. Лишь около одной трети владельцев смартфонов скачивают какие-либо приложения, и большинство из них загружают от одного до трех приложений в месяц. PWA способны изменить такую ситуацию.

Преимущества PWA-приложения:

1. Пользователи могут переходить на прогрессивные приложения из ссылок в соцсетях, во время просмотра веб-страниц или непосредственно из выдачи.

2. Предложение установить прогрессивное приложение показывается только тогда, когда веб-приложение отвечает определенным критериям, и пользователь продемонстрировал интерес посредством повторного посещения сайта.

3. Установка приложения происходит мгновенно. Все компоненты, которые требуют длительной загрузки, уже были установлены в кэш при первом посещении сайта пользователем.

4. Прогрессивные приложения гораздо меньше по размеру, так как они эффективно используют возможности браузера.

5. Всплывающие уведомления, работа в автономном режиме и все другие функции прогрессивного приложения будут работать, даже если посетитель никогда его не устанавливал.

6. Обычные приложения могут быть использованы только на той платформе, для которой они созданы. PWA же будут работать в любом месте, независимо от того, установлены они или нет.

7. Когда пользователь не в магазине приложений, то он не ограничен правилами App Store и не должен платить 30% от объема продаж.

Примеры результатов использования PWA-приложений внедрившими их компаниями:

Внедрив PWA, компания AliExpress увеличила коэффициент конверсии для новых пользователей на 104%.

United eXtra Electronics показал 4-кратное увеличение возвратов посетителей и на 100% повысил уровень продаж от пользователей, которые приходят в результате взаимодействия со всплывающими уведомлениями.

5miles снизил показатель отказов на 50% и увеличил конверсии на 30%.

Konga использует на 92% меньше данных для первоначальной загрузки по сравнению с загрузкой их native приложения.

Благодаря созданному в The Washington Post PWA количество посещений статей выросло на 12%. От загрузки статей по 8 секунд в 2013 г. они пришли к 80 миллисекундам в PWA. [4]

Что касается структуры Progressive Web Application, то можно сказать, что ее основу составляют:

1. Service Worker

2. Web App Manifests

3. Push notifications

Service Worker – это технология, которая отвечает за техническую часть и функционирование PWA.

Особенности работы благодаря Service Worker:

- Работа приложения в автономном режиме
- Повышение быстродействия из-за снижения количеств обращений к сети
- Удобные инструменты для обработки случаев проблем с соединением
- Синхронизация данных в фоновом режиме
- Централизованное получение обновлений результатов сложных вычислений для использования сразу несколькими частями приложения
- Повышение быстродействия за счет предзагрузки ресурсов

Технически Service Worker предоставляет в веб-браузере сценарий сетевого прокси для программного управления веб-запросами и HTTP-запросами. Service Workers лежат между сетью и устройством в дополнение к контенту. Они могут эффективно использовать механизмы кэширования и обеспечивать безошибочное поведение в автономном режиме.

Push- оповещения (нотификации, уведомления)

Push-уведомления позволяют отправить с сервера данные клиентскому PWA, даже когда оно может быть закрыто, и пользователь может не работать с ним.

Из Push Service Push-уведомления отправляются в Service Worker, который обеспечивает получение уведомлений и в том случае, если приложение закрыто. Так же, как и в обычных приложениях, нотификации происходят на системном уровне. Происходит запрос на разрешение от пользователей на отправку определенных уведомлений.

Оповещения должны быть персональные, должны приходить вовремя, быть уместными, краткими. Работа должна производиться вне зависимости от доступа к сети, оповещения не должны содержать рекламы.

Web App Manifest

Манифест веб-приложения - это спецификация W3C (World Wide Web Consortium), определяющая манифест JSON, чтобы предоставить разработчикам централизованное место для размещения метаданных, связанных с веб-приложением, включая:

- Наименование веб-приложения
- Ссылки на иконки веб-приложений или графические объекты
- Предпочтительный URL-адрес для запуска или открытия веб-приложения
- Данные конфигурации веб-приложения для ряда характеристик
- Объявление для ориентации веб-приложения по умолчанию
- Позволяет установить режим отображения, например, полноэкранный [5]

Вопросы безопасности

Приведем угрозы и методы решения, которые касаются нативных мобильных приложений и веб-сайтов.

Угрозы для мобильных приложений:

- Секретные данные в открытом виде;
- Небезопасные каналы передачи информации;
- Наличие отладочного кода;
- Внедрение SQL-операторов;
- Межсайтовый скриптинг (XSS);
- Отсутствие проверок входящих данных;
- Неправильная расстановка прав доступа;
- Слабая криптография.

К методам защиты относят:

- Обфускация кода
- Аутентификация

- Шифрование – средствами ОС, средствами приложения, средствами MDM
- Инструменты блокировки и\или очистки устройства при утере
- Аутентификация и контроль параметров устройства
- Аутентификация пользователя
- Ограничения:
    - По времени доступа к корп. ресурсам
    - По возможным точкам доступа [6]

Атаки и методы решения, касающиеся веб-сайтов

Атаки на веб-сайты:

- Подмена главной страницы сайта — одна из самых частых форм взлома.
- Удаление файловой системы — вся информация исчезнет, что становится особенно опасным в случае отсутствия сохраненной копии ресурса.
- Подмена информации — злоумышленники могут подменить телефон или другие данные организации.
- Размещение троянских программ — вредоносные программы могут выполнять разнообразные функции — осуществлять переадресацию на сайт злоумышленников, красть персональные данные клиентов, заражать вирусами.
- Рассылка спама — сайт могут использовать для рассылки спама, в этом случае «настоящая» корреспонденция не будет доходить до адресата, так как домен организации будет внесен в централизованную базу данных спамеров.
- Создание высокой нагрузки — отправление в адрес веб-сервера заведомо некорректных запросов или иные действия извне, результатом которых будет затрудненный доступ к сайту или падение операционной системы сервера.

Методы защиты веб-сайтов:

- Регулярно обновлять программное обеспечение - своевременное обновление программного обеспечения может помочь обезопасить сайт. Это относится как к серверному обеспечению, так и к любому обеспечению, которое может быть запущено на сайте.
- Всегда использовать параметризованные запросы, чтобы избежать SQL-инъекций.
- Создавать правильные сообщения об ошибках – нужно быть осторожным, когда даешь слишком много информации в сообщениях об ошибке. Нужно использовать общие сообщения типа «Неправильное имя пользователя или пароль». Не нужно уточнять, имя или пароль неверны, так как это позволит злоумышленнику понять, что он разгадал одно поле и может сконцентрироваться на другом.
- Проверки на стороне сервера, проверки в формах - проверка данных должны быть, как в браузере, так и на стороне сервера. В браузере можно, например, проверить на пустоту или на ввод только цифр. Однако, эти проверки могут быть пропущены и на сервер могут отправиться непроверенные данные.
- Нужно использовать комплексные пароли с буквами и цифрами.
- Безопасный сервер – рекомендуется установить firewall и блокировать все несущественные порты. Также установить DMZ (демилитаризованная зона), обеспечивающую доступ к порту 80 и 443. Для загрузки файлов на сервер, использовать только безопасные методы, такие как SFTP или SSH.
- Использование протокола SSL, который используется для обеспечения безопасности в Интернете. Каждый раз, когда передается информация между сайтом и web-сервером, используется сертификат безопасности. Если средства коммуникации не являются безопасными,

злоумышленники могут получить сертификат и доступ к данным пользователей [7].

Безопасность PWA-приложений

PWA работают с native API и Service worker, технологиями, которые связаны с конфиденциальными данными и должны обрабатываться с осторожностью. Поэтому каждый PWA должен подаваться через подключение HTTPS. Это обеспечивает безопасное соединение между сайтом и пользователями.

Благодаря использованию только протокола HTTPS обеспечивается защита пользователей от отслеживания на сайте. В противном случае злоумышленник мог бы использовать информацию о деятельности пользователя в браузере, включая поиск и другую персональную информацию.

Важно защитить свой бизнес от нежелательного контента. Угрозу могут представлять sql-инъекции, которые проникают на сайт без нашего ведома. Нежелательные объявления портят внешний вид сайта и ведут к доходному риску. Вредоносные программы подвергают риску пользователей сайта и могут вызвать серьезные поломки и разрушения.

Чтобы указать на небезопасное содержимое, в браузерах начинает меняться пользовательский интерфейс.

Безопасность также обеспечивается тем, что новые веб-функции будут доступны только с HTTPS протоколом. Браузер запретит использовать существующие и новые технологии, если они только с HTTP. [8]

Что касается практической части процесса изучения технологии Progressive Web Apps, то ведется разработка PWA-приложения для сайта S2W (StudyToWork) с функционалом для работы в нем студента, а именно, возможность просмотра студентом своего расписания и своих задач. Такое приложение позволит студенту всегда получать актуальное расписание, а

также следить за сроками выполнения имеющихся задач. Push-оповещения будут уведомлять о наступающем сроке сдачи задачи, об изменениях в расписании, приходе письма в чате от преподавателей, о возможности отметиться на мероприятии (лекция, практика), о поступившей новой задаче и т.д. Приложение предоставляет возможность открывать и просматривать расписание при отсутствии сети или при плохом соединении, так как оно способно работать в режиме offline. При внезапном разрыве соединения или при его отсутствии сообщения, отправленные преподавателю, обязательно достигнут своего назначения, они будут сразу же отправлены при появлении сети. Моменты безопасности, которые предусмотрены в PWA-приложении, позволят сохранить конфиденциальность данных студента, доступность приложения в любое время, целостность данных, отправляемых на проверку преподавателю и получаемых от него.

PWA-приложения взяли лучшие черты от мобильных приложений и веб-сайтов. Эта развивающаяся технология, полюбившаяся людям за ее удобство и расширенные функции, вносит большой вклад в развитие различных сфер. Взаимодействие пользователя с организациями становится еще более удобным, благодаря внедрению организациями таких приложений.

СПИСОК ЛИТЕРАТУРЫ

1. Bakaus P. What Are Progressive Web AMPs? [Электронный ресурс] // 2016. URL: https://www.smashingmagazine.com/2016/12/progressive-web-amps/ (дата обращения: 26.03.2017).
2. Dean A. H. Progressive Web Apps. N.Y.: Meap, 2017. P. 2-5.
3. Markov D. Progressive Web Apps. Everything You Should Know About Progressive Web Apps [Электронный ресурс] // 2016. URL: http://tutorialzine.com/2016/09/everything-you-should-know-about-progressive-web-apps/ (дата обращения: 26.03.2017).

4. Демьяненко М. Что такое Progressive Web Apps и какие возможности они открывают для вашего бизнеса [Электронный ресурс] // 2016. URL: https://netpeak.net/ru/blog/chto-takoe-progressive-web-apps-i-kakie-vozmozhnosti-oni-otkryvayut-dlya-vashego-biznesa/ (дата обращения: 30.03.2017).
5. Сальников М., Липатцев А., Кардава З., Пугачев С. Progressive Web Apps Day. Онлайн конференция [Электронный ресурс] // 2016. URL: pwaday.ru (дата обращения: 1.04.2017).
6. Мобильные приложения. Анализ защищенности мобильных приложений [Электронный ресурс] // 2016. URL: https://dsec.ru/services/security-analysis/mobile-applications/ (дата обращения: 3.04.2017).
7. 10 важных советов безопасности для защиты сайта [Электронный ресурс] // 2016. URL: http://alexdev.ru/1025/ (дата обращения: 3.04.2017).
8. Green I. From AMP to PWA [Электронный ресурс] // 2016. URL: https://www.slideshare.net/greenido/from-amp-to-pwa (дата обращения: 5.04.2017).