под критерии допустимости и качества, помещались в «белый список», а пользователи Adblock Plus видели их рекламу (если не отключали это в настройках).

Учитывая, как быстро происходит смена подходов к отображения рекламных объявления и нахождение способов борьбы с ней, современные способы блокирования рекламы являются неудовлетворительными. Сервисы не могут реагировать достаточно быстро на обратную связь пользователей, в следствие чего новая реклама появляется быстрее, чем блокируют старую.

Для обеспечения безопасности ментального здоровья следует разработать нейронную сеть, с возможностью настройки отображаемой рекламы. Система будет сама подстраиваться под предпочтения пользоваться, и отталкиваясь от них, будет блокировать ту или иную рекламу. Этот подход обеспечит своевременную реакцию на новые угрозы.

Автоматизировать распознавание рекламы сложно, помимо прочего, еще и потому, что даже у людей нет единого мнения насчет того, что является рекламой, а что нет. Поэтому приложение будет отслеживать поведение конкретного пользователя на странице.

После установки приложения пользователю предлагается выбрать определенный паттерн блокировки рекламы. В него будут включаться сайты из «белого списка», а также наиболее приемлемые паттерны рекламы по отзывам других пользователей.

Основываясь на общей статистике, приложение будет скрывать ту рекламу, которая оказалась наиболее неприемлемой, по мнению других пользователей. Рейтинг неодобрения пользователей будет основываться на использовании пользователями одного из способов защиты от негативного информационного влияния – «ухода». Приложение будет отслеживать положение курсора, клики и окна, которые будут активно использоваться. Будет отслеживаться насколько долго реклама находилась на странице, пока пользователь ее не зарыл. Чем это время больше – тем выше лояльность пользователей к данному типу рекламы.

Также активно будет использоваться технология аиртрекинга [3]. Она будет отслеживать на сколько долго пользователь фокусировал внимание на блоках рекламы. Также, как и в случаях с пользовательским вводом, чем выше это время – тем выше лояльность пользователей.

Основываясь на этих данных, приложение будет скрывать «неугодную» для пользователя рекламу. Однако полностью скрыть всю рекламу не представляется возможным. Это происходит потому, что реклама является, в большинстве своем, основным источников доходов для владельцев сайта и крупных поисковых систем.

Постепенно накапливая и анализируя информацию, будут приниматься соответствующие меры по блокированию рекламы.

Список использованных источников:

1. Дроздова, А. В. Воздействие рекламы на безопасность личности в современном информационном обществе: социально-психологический аспект /А. В. Дроздова // Вестник Московского университета. Серия 14. Психология – Москва: МГУ, 2011. – с. 58-65.

2. Филиппова, Т.В. Интернет как инструмент социологического исследования / Т.В. Филиппова, Т.В.Дроздова // Социологические исследования. Выпуск 4 – Красноярск: КГАУ, 2001. – с. 115-122.

3. Окулография. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Окулография. – Дата доступа: 20.03.2018.

ЗАЩИТА ОТ АВТОМАТИЧЕСКОГО ТРАФИКА

Институт информационных технологий БГУИР,
г.Минск, Республика Беларусь

Казанок Д.Ю., Новохрестова А.О.

Савенко А.Г.- ассистент каф ПЭ, м.т.н.

При построении информационных систем различного уровня сложности нередко является актуальным вопрос защиты сервиса от различного рода нежелательных внешних воздействий. Отдельно хотелось бы остановиться на мобильных- и веб-приложениях. В обоих случаях они могут быть тонкими клиентами над серверными приложениями, работающими в облаке. Соответственно, при разработке таких приложений возникает вопрос безопасности серверных приложений против различных форм автоматических запросов, направленных через или в обход разработанного клиентского приложения.

Отсутствие защиты от автоматического трафика при его регулярном поступлении приводит к затратам рабочего времени, оплачиваемого работодателем, на ликвидацию нанесенного ущерба. Так же автоматический трафик значительно увеличивает нагрузку на коммуникации и снижает эффективность работы сервера. Резюмируемый итог: дополнительные финансовые расходы и угроза целостности пользовательских данных (в том числе, при устранении последствий).

Для решения такого рода задач существует понятие CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). Как следует из названия, задача состоит в том, чтобы отличить роботов от человека во время работы приложения [1]. Заметим, что в мобильных приложениях такая задача обретает свою специфику реализации и влияния на пользовательский опыт, что, в свою очередь, оказывает влияние на выбор того или иного способа реализации. Приведем самые актуальные варианты реализации CAPTCHA и особенности их интеграции:

1) Google reCAPTCHA V2

При этом способе анализируются показатели движения мыши, траектория ее движения и отклонения, а также наличие человеческой активности в файлах cookie. Отслеживание данных пользователя покажет, человек

он или нет, и только тогда предложит отметить это галочкой в поле «Я не робот». После этого — все, готово. Однако, если алгоритм Google reCAPTCHA V2 не смог точно определить человека, то следующим этапом будет показано изображение, скажем, кошки. Затем панель с изображениями под картинкой кошки. Нужно будет найти все картинки, связанные с кошкой (котята, тигры, львы).

Google reCAPTCHA V2 является одним из самых широко используемых вариантов защиты от автоматического трафика в веб-приложениях за счет простоты прохождения теста пользователем и высокого процента определения правдивого результата. Для веб-приложений процесс интеграции состоит в следующих действиях: подключить скрипт reCAPTCHA API к странице, зарегистрировать домен сайта в административной панели при помощи аккаунта Google, вложить в форму div-элемент с классом "g-recaptcha", где в атрибуте "data-sitekey" указать ключ, полученный в административной панели [2].

В мобильных приложениях интеграция сопряжена с трудностями, т.к. потребуется показывать веб-содержимое, а веб-контейнеры в мобильных приложениях не содержат домена, совместимого с reCAPTCHA, что вынуждает хранить данное веб-содержимое на специально приобретаемом для этого веб-хостинге.

2) Invisible reCAPTCHA

Суть новой версии reCAPTCHA от Google в том, что теперь не будет необходимости куда-либо нажимать/вводить/выбирать, чтобы подтвердить, что вы человек. Для пользователей всё будет прозрачно, если у CAPTCHA не будет сомнений в его «человечности».

В новой версии системы сочетаются машинное обучение и продвинутый инструмент для анализа риска, который адаптируется к новым угрозам.

Более подробная информация о системе, вероятно, поможет создателям ботов придумать способ обойти её, поэтому в ближайшее время деталей можно не ждать.

Процесс интеграции в веб-приложениях на данный момент аналогичен процессу интеграции reCAPTCHA V2. Единственное отличие в том, что на новый элемент div нужно добавить атрибут "data-size" со значением "invisible" [2].

Использование Invisible reCAPTCHA в случае мобильных устройств представляется возможным только в гибридных приложениях, где CAPTCHA получит возможность анализировать действия пользователя. В то же время данная технология наследует все трудности реализации в мобильных приложениях от reCAPTCHA V2.

3) Взаимодоверяющая CAPTCHA

Еще один вид распознавания человека на основе изображения. Он дает до 96% точности, что очень неплохо, к тому же работает в два раза быстрее, чем классическая CAPTCHA. Взаимодоверяющая CAPTCHA действует по такому принципу: показывает несколько пар картинок и затем задает некоторое количество вопросов об этих картинках. Пользователь отвечает, кликая мышкой по нужным изображениям.

Стоит отметить, что данный вид CAPTCHA в совокупности с оригинальным дизайном может стать не просто оригинальным способом проверки, превращенным в мини-игру, но и гармонично вписаться в него, не ухудшая внешний вид приложения.

Готовых решений, применимых в веб- и мобильных приложениях, которые можно было бы широко рекомендовать, как в случае с решением от компании Google, нет. В основном, такой вариант CAPTCHA проектируется разработчиком под конкретное приложение. Общий случай реализации представляет собой следующие действия: создается база данных, содержащая изображения и метки, соответствующие этим изображениям; каждой метке ставится в соответствие вопрос, который будет выводиться пользователю; веб-сервер передает клиенту вопрос, и несколько изображений, имеющих общую метку, и одно или более изображение с иной меткой. В этот момент веб-сервер ассоциирует метку вопроса с пользовательской сессией (хранить в неявном виде) для дальнейшего извлечения. После того, как пользователь выбрал ответы, на сервер отправляются данные о не выбранных вариантах ответа, чтобы обрабатывать меньше данных на сервере. Остается найти соответствующие переданным изображениям метки, привести их в одинаковый вид с хранимым значением сессии пользователя и сравнить их. Если будет найдено хотя бы одно совпадение - тест не пройден [3].

Данный способ одинаково применим и для мобильных, и для веб-приложений.

Резюмируя вышесказанное, компания Google предоставляет наиболее удачное решение для предотвращения автоматического трафика веб-приложений (reCAPTCHA). В мобильных приложениях для устройств с операционной системой iOS интеграция потребует дополнительных ухищрений: наличие хостинга и доменного имени, а также использование инструментов отображения веб-содержимого. В мобильных приложениях для устройств с операционной системой Android, компания Google предоставляет API, что позволяет комфортно интегрировать reCAPTCHA. Однако в случае, когда использование Google reCaptcha не является целесообразным по ряду причин, имеет смысл затратить дополнительные ресурсы на разработку взаимодоверяющей CAPTCHA (например, в мобильном приложении для устройств с операционной системой iOS).

В приложениях, требующих повышенной безопасности данных пользователя (например, приложения осуществляющие операции с денежными средствами), следует рассмотреть вариант отказа от CAPTCHA в пользу использования привязки к номеру мобильного телефона.

Верификация мобильного номера происходит следующим образом: пользователь вводит свой мобильный номер в специальную форму, система генерирует пароль и отправляет его по SMS на указанный телефонный номер. Пользователь должен получить SMS с кодом и ввести его в предназначенное поле, чтобы подтвердить свою личность. Если введенный код совпадает с тем, что был отправлен системой, то пользователь получает разрешение на совершение дальнейших действий в приложении.

Таким образом, снижается уровень нежелательного автоматического трафика при условии использования дополнительных мер по его выявлению. Этими дополнительными мерами могут являться: введение ограничений на количество запросов с одного номера в определенный период времени; введение системы подачи и

мониторинга пользовательских жалоб. Набор таких мер является более эффективным в ряде случаев, чем CAPTCHA, и применяется компаниями-гигантами в сервисах охватывающих огромную долю рынка.

Список использованных источников:
1. Bursztein, E. How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation / E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, D. Jurafsky. - Stanford, 2010. - 15 с
2. Google Developers [Электронный ресурс] / Support and resources to develop. – Режим доступа: https://developers.google.com. – Дата доступа: 11.03.2018.
3. Chew, M. Image Recognition CAPTCHAs / M. Chew, J. D. Tygar. - Berkeley, 2004. - 19 с.

ВИРТУАЛИЗАЦИЯ СЕРВЕРОВ С ПОМОЩЬЮ WEBSPHERE

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Кармызов А.С.

Скудняков Ю.А.- доцент каф. ПЭ, к.т.н,, доцент

В данной работе рассматриваются возможности виртуализации серверов информационно-вычислительной системы с использованием программно-аппаратного обеспечения WebSphere для достижения оптимальных значений таких показателей качества серверов как: максимальное использование их ресурсов, оперативное предоставление услуг, снижение капитальных затрат и уменьшение административных расходов.

Виртуализация серверов предоставляет множество преимуществ, касающихся многих различных типов сред, включая среды прикладных систем промежуточного программного обеспечения [1]. В первую очередь виртуализация серверов позволяет достигать более высокого коэффициента использования ресурсов, тем самым помогая исключить серьезное недоиспользование серверных ресурсов, характерное для столь многих организаций. Работа нескольких экземпляров программного стека на одной машине часто обеспечивает более полное использование ресурсов сервера по сравнению с ситуацией, когда на сервере работает всего один программный стек. Помимо повышения коэффициента использования ресурсов, виртуализация серверов также может обеспечивать ускоренное предоставление услуг, снижение капитальных затрат и уменьшение административных расходов. Однако, чтобы получать отдачу от этих и любых иных преимуществ виртуализации серверов, необходимо эффективно осуществлять управление ее использованием.

Программно-аппаратное решение IBM WebSphere CloudBurst™ обеспечивает возможность эффективного управления виртуализацией серверов для сред IBM WebSphere Application Server. Как показано на рисунке 1, данная система ориентирована на полный жизненный цикл среды Web-приложений в виртуализованной среде.

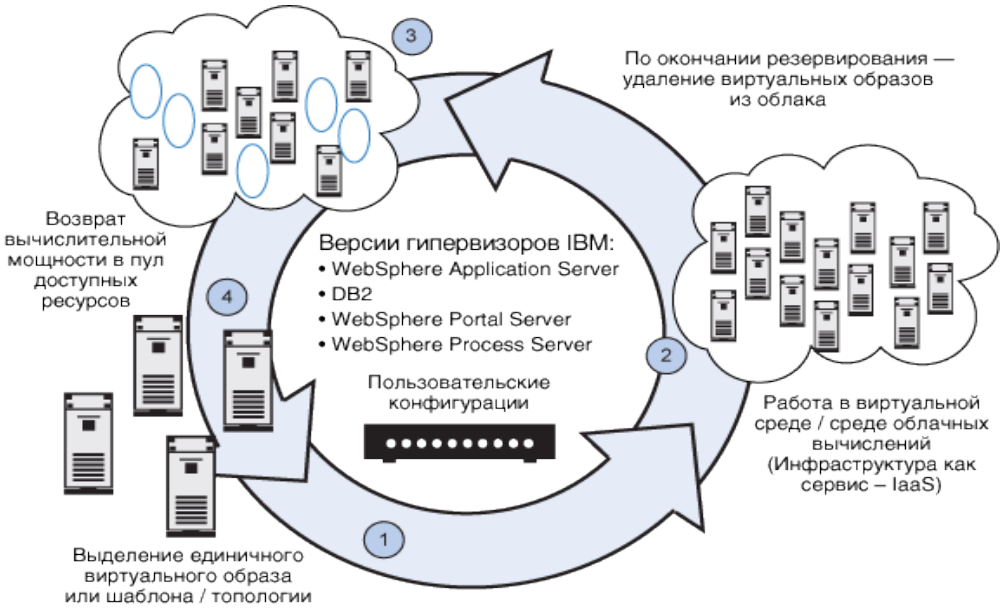

Рисунок 1 – Эффективное управление виртуализацией серверов с помощью WebSphere

CloudBurst начинает работу с того, что дает возможность определить пользовательские среды, охватывающие от одной до нескольких виртуальных машин. Затем решение осуществляет автоматизацию и оркестровку развертывания и конфигурирования виртуальных машин для создания адаптированной среды WebSphere Application Server, работающей на виртуализованных серверах. После развертывания можно использовать решение для централизованного управления и мониторинга виртуализованной среды. Когда