УДК 37.02:004

О. Ю. САМОЛЮК, В. В. МЕЛЕНЧУК
УО «Брестский государственный технический университет» (г. Брест, Беларусь)

ЭВОЛЮЦИЯ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ САРТСНА В СОВРЕМЕННОМ МИРЕ

В Интернете сейчас огромное количество пользователей. Однако далеко не все из них являются реальными людьми, в сети можно встретить множество различных ботов. Бот – программа, выполняющая повторяющиеся заранее настроенные автоматические задачи [1]. Зачем нужны боты? Они позволяют пользователю выполнять какие-либо задачи без его участия, облегчая его работу. Разумеется, ничто не мешает пользователю иметь более одного бота, что создаёт риск того, что этот пользователь будет создавать иллюзию трафика на сайте или вообще попытается вывести его из строя, перезагрузив его запросами.

Для борьбы с подобными проблемами и была создана капча. Говоря простыми словами, капча – это тест для проверки того, является ли пользователь сайта или приложения человеком. Капча создаётся таким образом, что у человека не должно возникать проблем при выполнении сгенерированного задания, в то время как для бота эта задача должна быть нерешаема.

Разберём наиболее популярные виды капчи. Первым основным типом капчи является так называемая капча подтверждения действия. От пользователя просто требуется подтвердить, что он не робот. Это самый часто встречающийся вид капчи ввиду того, что он наипростейший в разработке. От разработчика не требуется ничего,

кроме добавления на экран дополнительной кнопки или поля, в котором просто нужно поставить галочку для продолжения. И так как этот вид капчи является самым простым для добавления, он является и самым ненадёжным, так как боты могут пройти эту проверку без серьёзных затруднений, если встроить в них функцию перебора нажатия по случайным точкам экрана в случае, если запросы перестали обрабатываться.

Вторым типом капчи является текстовая капча. Пользователю необходимо ввести текст с картинки в специальное поле для ввода. Этот тип капчи уже требует определённых знаний для разработки. В качестве основы для данной капчи можно взять как пустую картинку, на которой через определённые функции языка можно нарисовать текст, так и просто отображаемое текстовое поле. Так как второй вариант более простой и гибкий в настройке, рассматривать будем его. В первую очередь необходимо составить алфавит используемых символов, обычно используются все буквы английского алфавита плюс цифры десятичной системы счисления. Разработчик может удалить из алфавита нежелательные символы, которые будет сложно распознать и человеку (например, букву ‘o’, которая может быть очень похожа на цифру ‘0’). Обычно такая капча состоит из 4–6 символов, поэтому разработчику необходимо будет при каждом показе капчи выбирать из алфавита необходимое количество символов случайным образом. Выбранная последовательность символов должна быть запомнена программой для последующей проверки вводимого текста. Также разработчику необходимо добавить фон и шум для полученного текста. Фон должен быть приближен к цвету текста для усложнения распознавания, но не должен сливаться с ним. Шум – это обычно линии и точки, которые накладываются на текст для усложнения распознавания, их рекомендуется делать того же цвета, что и сам текст капчи. К тому же, рекомендуется деформировать сами буквы различными способами для достижения максимальной нечитабельности, но необходимо понимать, что тут важно не перестараться, иначе даже человеку окажется не под силу распознать сгенерированную строку. Данный тип капчи хорошо работает против примитивных ботов, но более сложные версии могут быть оснащены алгоритмом распознавания текста (с примером нейронной сети, работающей по данному алгоритму можно ознакомиться в [2]). Чем более продвинутым будет алгоритм, тем более сложные версии текстовой капчи он сможет распознать.

Следующим типом капчи, с которым сталкивался почти каждый пользователь сети Интернет, является капча с изображениями. Пользователю показывают картинку, на которой просят найти фрагменты с какими-то конкретными объектами (например, со светофорами) либо набор из картинок, в котором ему необходимо выбрать лишь те картинки, которые подходят под заданное условие. Данный тип капчи более надёжный, чем текстовая, однако это компенсируется гораздо более сложным процессом разработки. В случае цельной картинки разработчику необходимо создать массив из кнопок, после чего разрезать изображение на части и «наклеить» на каждую кнопку соответствующую часть, а затем отметить, какие кнопки должны быть нажаты, а какие – нет. В случае с набором картинок необходимо «клеить» целые картинки, предварительно задав теги для картинки, чтобы программа могла определить, для каких заданий данная картинка является корректным ответом. Также важно учитывать вероятность человеческой ошибки, проще говоря, человек может ошибиться и выбрать неправильный вариант, поэтому необходимо считать корректным результатом не полностью правильное решение, а допускать несколько неправильных ответов. К сожалению, технологии дошли до такого уровня развития, что боты вполне способны решать и такие задачи (с примером подобного алгоритма можно ознакомиться в [3]), поэтому наличие капчи с изображениями не гарантирует безопасность.

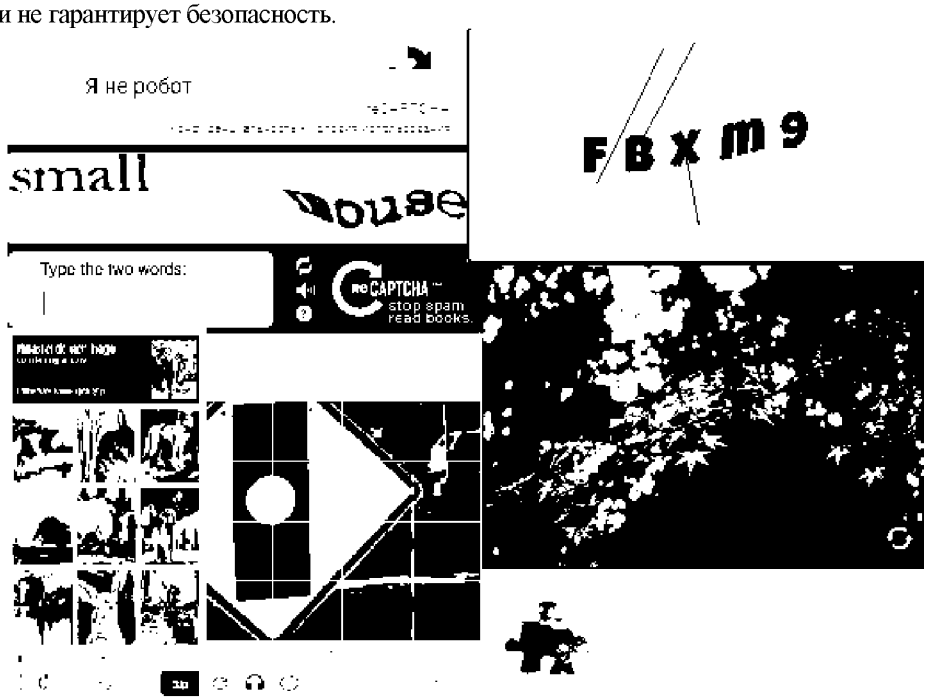

Рисунок 1 – Компиляция основных видов капчи

Также можно отметить капчи с задачами. Вас могут попросить решить простенький пример или ответить на несложный вопрос. Данный тип капчи является малоэффективным, так как боту достаточно распознать вопрос, попробовать ввести его в поле для ввода и в случае ошибки сделать запрос в поисковую систему, после чего ввести уже ответ. К тому же подобные капчи либо используют заранее заданный набор вопросов, либо требуют постоянного обновления набора заданий, либо достаточно сложны в разработке, что ведёт к их низкой популярности.

Последним распространённым типом капчи являются капчи-пазлы. Суть этих капч сводится к тому, что пользователю необходимо перетянуть элемент на нужное место. Однако даже такие капчи на сегодняшний день легко ломаются (с примером того, как решают подобные задачи с помощью кода, можно ознакомиться в [4]).

Из редких видов капчи можно выделить голосовую капчу [5]. Её суть заключается в том, что пользователю необходимо произвести голосовую запись для подтверждения того, что он – человек. Но даже такую капчу можно обойти (например, с помощью синтеза голоса), а в плане разработки данная капча является наиболее сложной, требующей наибольшего количества времени. К тому же данная капча не слишком удобна для пользователя (например, у него может не быть микрофона в компьютере), что тоже не способствует её популярности.

Разумеется, существует множество готовых решений и библиотек, которые можно включить в свой проект, получив возможность добавить созданную другими разработчиками капчу без лишних трудозатрат. Однако данный способ не гарантирует необходимую гибкость и возможность устранения проблем с безопасностью капчи в случае их возникновения.

Как следует из статьи, на сегодняшний момент не существует гарантированного способа защититься от ботов с помощью капчи. Технологии дошли до такого уровня, что между программистами идёт бесконечная борьба: одни придумывают сложную для обхода капчу, вторые – способ её обхода. Создаются различные алгоритмы для решения определённых шаблонов капчи, нейронные сети, которых учат распознавать необходимые картинки по выборке изображений и многое другое (подробнее можно почитать в [6]). Необходимо понимать, что не существует способа защититься от ботов на 100%, любую защиту можно обойти и в таком случае важна оперативность реакции разработчика на обход. К тому же, большинство злоумышленников будет использовать примитивных ботов, для которых испытанием может стать само наличие капчи. Поэтому всегда важно находить золотую середину между гибким и надёжным решением и затратами времени и ресурсов на его реализацию.

Список использованных источников

1. Что такое боты – определение и описание [Электронный ресурс]. – Режим доступа: https://www.kaspersky.ru/resource-center/definitions/what-are-bots/. – Дата доступа: 05.02.2024.

2. Редуцированная сверточная нейронная сеть для точного распознавания рукописных цифр / В. А. Головко // Вестник Брестского государственного технического университета. Серия: Физика, математика, информатика. – 2016. – № 5. – С. 1–7.

3. Глубокое обучение для детектирования объектов на изображениях документа / А. А. Крощенко // Вестник Брестского государственного технического университета. Серия: Физика, математика, информатика. – 2017. – № 5. – С. 1–9.

4. Как решать капчи-слайдеры от GeeTest с помощью JS [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/508690/. – Дата доступа: 05.02.2024.

5. Голосовая капча: эффективное решение для защиты от ботов [Электронный ресурс]. – Режим доступа: https://vc.ru/marketing/741492-golosovaya-kapcha-effektivnoe-reshenie-dlya-zashchity-ot-botov. – Дата доступа: 05.02.2024.

6. Brodic, D. The CAPTCHA: Perspectives and Challenges, Smart Innovation, Systems and Technologies, September 18, 2019 / D. Brodic, A. Amelio. – Springer, 2019. – Vol. 162. – P. 105–118.