УДК 004.056.53 **DOI: 10.18413/2518-1092-2020-5-3-0-3**

Девицына С.Н.
Гоголь А.С.

РАЗРАБОТКА МЕТОДА СОЗДАНИЯ КАПЧИ, УСТОЙЧИВОЙ К АВТОМАТИЧЕСКОМУ РАСПОЗНАВАНИЮ И УГАДЫВАНИЮ

Севастопольский государственный университет, ул. Университетская, д. 33, г. Севастополь, 299053, Россия

e-mail: sndevitsyna@sevsu.ru, andrewgogol777@gmail.com

Аннотация
Актуальность рассматриваемой в статье проблемы обусловлена тем, что растет количество веб-ресурсов различных государственных организаций и коммерческих компаний в сети Интернет, а одной из причин утечки данных может являться массовый сбор информации. Полученные при сборе сведения могут стать инструментом в умелых руках злоумышленника. Злоумышленники могут спокойно создавать и выгружать в сеть вредоносное программное обеспечение с функциями сбора информации, собранная информация может использоваться для осуществления атак с мощью методов социальной инженерии. Самыми подходящими методами для таких атак являются фишинг и претекстинг. В статье предлагается обзор проблемы незаконного массового сбора информации и использование её злоумышленниками. Проанализированы возможные методы противодействия массовому сбору информации, рассмотрены варианты создания капч, и их недостатки – возможность распознавания и угадывания злоумышленником, или ботом. В результате предложен улучшенный метод, который решает данную проблему. В работе описаны основные функции работы программы, а также возможные вариации использования генерации капчи. Для защиты от распознавания капчи обученным ботом, предложено вводить в изображение смысловую нагрузку. В результате разработан и представлен метод создания капчи, устойчивой к автоматическому распознаванию текста.
Ключевые слова: информационные технологии; информационная безопасность; капча; авторизация; сбор информации; распознавание образов.

UDC 004.056.53

Devitsyna S.N
Gogol A.S.

DEVELOPING A METHOD FOR CREATING A RESISTANT TO AUTOMATIC RECOGNITION AND GUESSING CAPTCHA

Sevastopol state University, 33 Universitetskaya St., Sevastopol, 299053, Russia

e-mail: sndevitsyna@sevsu.ru, andrewgogol777@gmail.com

Abstract
The relevance of the problem considered in the article is due to the growing number of web resources of various government organizations and commercial companies on the Internet, and one of the reasons for data leakage may be mass collection of information. The information obtained during the collection can become a tool in the capable hands of an attacker. Attackers can easily create and upload malicious software with information collection functions to the network, and the collected information can be used to carry out attacks with the power of social engineering methods. The most suitable methods for such attacks are phishing and pretexting. The article provides an overview of the problem of illegal mass collection of information and its use by hackers. Possible methods of countering mass data collection are analyzed, options for creating captchas are considered, and their disadvantages are the possibility of recognition and guessing by an attacker or bot. As a result, an improved method is proposed that solves this problem. This paper describes the main functions of the program, as well as possible variations in the use of captcha generation. To protect against captcha recognition by a trained bot, it is suggested to enter a semantic load into the image. As a result, we developed and presented a method for creating a captcha that is resistant to automatic text recognition.

Keywords: information technology; information security; CAPTCHA; authorization; information collection; image recognition.

ВВЕДЕНИЕ

Применение методик социальной инженерии приводит к тому, что сотрудник компании разглашает сведения, которые несут определенную ценность для организации, в которой он работает. Данные методики могут использоваться также и для осуществления атак на рядовых пользователей, например, для сбора сведений о банковском счете, либо банковской карте.

Для защиты веб-ресурсов от массового сбора информации пользователей, которая потом может использоваться злоумышленниками, администратор и разработчики сайта могут использовать защитное программное обеспечение. Некоторые из применяемых средств имеют недостатки, которые были учтены и удалены при разработке программы для защиты сайта от ботов.

Действенным инструментом злоумышленников являются сети из зараженных вирусами компьютеров. Данные сети называют ботнетами [Kaspersky, 2020], они используют чужие вычислительные мощности, а также занимаются саморасширением, и запрограммированы на какие-либо повторяющиеся действия. В эти повторяющиеся действия может входить и сбор информации. Для человека это – очень длительный процесс, в то время как бот справится с ним намного быстрее. Предлагаемый программный продукт будет включать в себя и предупреждения для людей, чьи компьютеры оказались частью сети злоумышленника.

Целью исследования является улучшение существующих методов защиты сайтов от массового сбора информации. Для достижения поставленной цели решены следующие задачи:

- проанализированы существующие меры защиты сайтов от массового сбора информации;
- разработан алгоритм действий для защиты сайтов от массового сбора информации.

ОСНОВНАЯ ЧАСТЬ

С развитием информационных технологий Тест Тьюринга [Turing test, 2020] нашел свое применение в повседневной работе сайтов и защите их от сбора информации. Для защиты от массового сбора информации часто используют тесты, такие, как капча.

Капча – тест, который является модификацией теста Тьюринга и служит для распознавания и отсеивания ботов с веб-ресурсов [Habr, 2009; reCAPTCHA 2020; Habr, 2011].

Различают следующие виды тестов:

- интерактивные тесты;
- иллюстративные тексты;
- тесты на логику;
- тесты со смысловой нагрузкой;
- текстово-цифровые тесты.

Все виды капч имеют свои особенности [CAPTCHA, 2020; reCAPTCHA 2020; Rucaptcha. 2020]. Например, чтобы пройти интерактивный тест, необходимо взаимодействовать с интерфейсом сайта, и пользователя могут попросить передвинуть ползунок в необходимое положение. Чтобы пройти капчу со смысловой нагрузкой, пользователя могут попросить решить легкую загадку, либо проверить его на внимательность. В случае теста на логику можно попросить пользователя решить математический пример: сложить 1 и 2. Каждая такая программа должна использоваться администратором на основании каких-либо признаков, либо может работать в автоматическом режиме.

Самыми слабыми капчами являются текстово-цифровые и логические [Пудовикова П.Д., Титов С.С., 2016]. Логические можно перебрать, а текстово-цифровые можно распознать. При разработке метода создания капчи необходимо не только учесть недостатки существующих видов капчи, но и то, что существуют группы лиц, которые работают над распознаванием капч. Таким образом, метод должен позволить отсеивать определенную часть таких злоумышленников.

Злоумышленники могут использовать базы данных, которые содержат в себе миллионы ответов на разные капчи. Чтобы избавиться от этой проблемы, необходимо улучшить количество вариаций при генерации изображения капчи. Признаками взлома могут являться: необычное поведение пользователя, повторяющиеся безрезультативные действия, использование и переходы по скрытым ссылкам, либо странные движения мышью.

Надежный алгоритм теста должен обладать такими свойствами как:

- устойчивость к распознаванию;
- защита от перебора;
- устойчивость к угадыванию.

При выборе вида капчи рассматривались недостатки существующих методов. Например, на рис. 1 представлена слабая капча с фиксированным и неискаженным шрифтом. Такой капче свойственны легкость отделения текста с помощью цветового ключа и легкость отделения символов друг от друга.

На рисунке видно отделение символов, которые темнее определенного уровня, таким образом работает автоматическое распознавание текста. Весь остальной фон заполняется белым цветом.

Рис. 1. Пример разбора слабой капчи
Fig. 1. Example of parsing a weak captcha

На рис. 2 представлен пример сильной капчи – reCAPTCHA от компании Google [reCAPTCHA 2020]. Для ее генерации компания Google использует слова, которые не смогли распознать их боты. Для большей надежности используется проверочное слово, данное слово намеренно искажается. В этом случае рекомендуется дублировать текст, либо использовать горизонтальное и вертикальное искажение.

В капче компании Google слова берутся из старых учебников, которые оцифровали для перевода в электронный формат и хранят на серверах компании. Для большей точности предлагается пройти тест с одним и тем же словом тысячам пользователей.

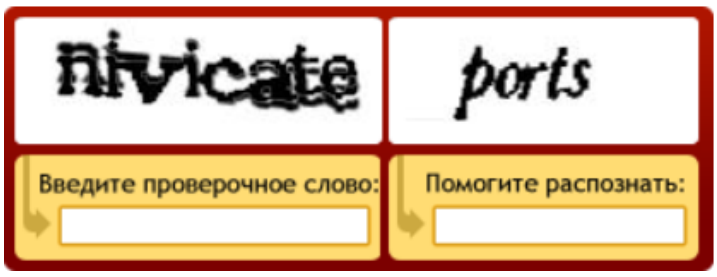

Рис. 2. Пример сильной reCAPTCHA от компании Google
Fig. 2. Example of a strong reCAPTCHA from Google

Проанализировав сильные и слабые стороны разных видов капчи, принято решение предложить альтернативный вариант, позволяющий защитить капчу от автоматического распознавания и угадывания. Для разработки приложения были выбран язык Python [Sololearn Python, 2020].

На рис. 3 представлена структурная схема алгоритма работы программы, которая включает в себя следующие функции:

- функция создания матрицы с данными для демонстрации работы с пользователями;
- функция генерации случайной строки из букв и цифр;
- функция нанесения текста на изображение;
- функция цветовой инверсии;
- функция проверки пользователя.

Рис. 3. Структурная схема алгоритма программы
Fig. 3. Block diagram of the program algorithm

Рассмотрим три функции, которые определяют устойчивость разработанного метода. Это – функции генерации случайной строки, функция нанесения текста на изображения, а также функция цветовой инверсии. Все эти функции связаны принципами функционального программирования и создают устойчивую к распознаванию капчу. В каждую из последующих функций заходит значение, которое возвращает предыдущая функция.

Основными инструментами создания устойчивой к автоматическому распознаванию капчи являются: функция генерации капчи, нанесения капчи на изображение из базы данных изображений и функция цветовой инверсии.

На рис. 4 приведен сгенерированный случайный текст, состоящий из букв кириллицы и цифр. Использование кириллических букв позволит частично отсеять сегмент иностранных злоумышленников, выполняющих распознавание капч.

Функция работает с заранее прописанной строкой, в которой имеются цифры от 0 до 9 и буквы кириллицы. Далее, с помощью библиотеки random и определенного метода, выбирается случайное количество символов в пределах от 7 до 9 из заранее определенной строки. Данная функция сохраняет результат своей работы в список, первый и единственный элемент которого – это выбранные символы.

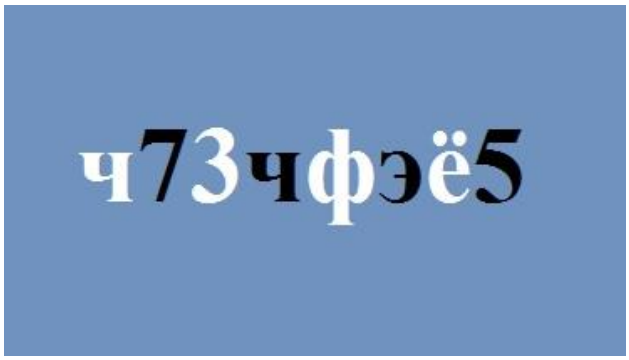

Рис. 4. Сгенерированный текст
Fig. 4. Generated text

На рис.4 видно, что текст уже перекрашен в черно-белый цвет, но данное преобразование должна выполнять функция нанесения текста на изображение с помощью использования библиотеки PIL. Работа с данной библиотекой возможна при использовании среды разработки PyCharm и установленного пакета библиотек Anaconda.

PyCharm – это интегрированная среда разработки (IDE), используемая в компьютерном программировании, особенно часто для языка Python. Она разработана чешской компанией JetBrains. PyCharm включает в себя анализ кода, графический отладчик, встроенный тестер модулей, интеграцию с системами контроля версий (VCSes), и поддерживает веб-разработку с Django, а также DataScience с Anaconda [Anaconda, 2020; Pycharm, 2020].

Anaconda – дистрибутив языков программирования Python и R, включет набор популярных бесплатных библиотек, объединенных проблемами науки о данных и машинного обучения. Основная цель – предоставить тематическим модулям единый согласованный набор наиболее востребованных соответствующим кругом пользователей для разрешения возникающих зависимостей и конфликтов, которые неизбежны при одной установке [Sololearn Python, 2020.].

На рис. 5 представлен пример работы функции нанесения символов на изображение с помощью методов DrawText. При нанесении создаются два списка разных цветов с пробелами, что позволяет нанести символы черно-белого цвета в том порядке, в котором они были сгенерированы.

Рис. 5. Нанесение сгенерированного текста на выбранное изображение
Fig. 5. Applying the generated text to the selected image

На рис. 6 представлен результат работы функции инверсии изображения, которое получено из функции нанесения текста на изображение. Именно это изображение будет представлено боту, либо человеку. Пользователя попросят ввести символы, например, белого цвета, расположенные в верхней части изображения. Можно предложить выбрать символы черного цвета, либо комбинировать области изменения цвета, либо просить его ввести символы того или иного цвета из верхней части изображения. Комбинации могут быть различными, главное – наличие

смысловой нагрузки и обеспечение выбора заданных типов символов для прохождения аутентификации.

Рис. 6. Пример готовой капчи с инверсией пикселей
Fig.6. Example of a ready-made captcha with pixel inversion

Помимо всего, необходимо предупреждать пользователя, что его компьютер может являться частью ботнета, что, возможно, поможет исключить один бот из сети злоумышленника. В программе учитывался данный фактор, так как перед тестом мог находиться пользователь, чей компьютер нес в себе вредоносное программное обеспечение, а не автономный бот, который использует ресурсы злоумышленника, либо сервисы для сканирования сайтов.

В результате работы была получена капча, устойчивая к автоматическому распознаванию текста. Так как не исключается возможность распознавания данной капчи обученным ботом, в ней присутствует смысловая нагрузка, а именно – просьба ввести определенные символы с той или иной области изображения.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

В результате исследования был выявлен ряд недостатков во всех видах существующих тестов. Для создания устойчивой капчи предложено комбинировать методы из каждого типа капч. При создании устойчивого к автоматическому распознаванию и угадыванию метода была использована комбинация двух видов капч: текстово-цифровая капча и капча со смысловой нагрузкой. Смысловая нагрузка является страховкой метода, которая защищает его от автоматического распознавания текста. Также были учтены физиологические особенности человека: так как не каждый человек может различать цвета, предложено использовать простую черно-белую гамму. Также плюсы данной капчи в том, что можно выбирать тематику изображения, что позволит использовать ее на сайтах различных государственных организаций и компаний. Количество вводов и время ввода может выставлять сам администратор сайта. Данный программный продукт, написанный на языке программирования Python, может использоваться большим количеством веб-ресурсов, так как основным их языком является язык Python.

ЗАКЛЮЧЕНИЕ

Сегодня у злоумышленников имеется огромное количество ресурсов в виде программ-скрапперов в составе ботнета, автономных ботов и общедоступных интернет-утилит. Преступники могут собирать информацию не только сами, но и с помощью чужих ресурсов, что увеличивает масштабы сканирования и сбора информации. Следовательно, это влечет за собой определенные угрозы для рядовых пользователей и сотрудников организаций.

В статье показан улучшенный метод создания капчи, предложенный Гоголем А.С. в рамках выпускной квалификационной работы бакалавра, также представлены функции и алгоритм программы для защиты сайтов от массового сбора информации. К достоинствам улучшенного алгоритма, написанного на языке Python с использованием среды разработки PyCharm, можно отнести:

- устойчивость к распознаванию текста;
- наличие смысловой нагрузки;
- простая цветовая гамма;
- наличие предупреждения для пользователя.

Данную капчу можно использовать как при работе пользователя на сайте, так и для процедуры авторизации.

Список литературы

1. Anaconda, 2020. URL: https://www.anaconda.com (дата обращения 05.05.2020).
2. CAPTCHA: Telling Humans and Computers Apart Automatically. URL: http://captcha.net (дата обращения 03.05.2020).
3. Habr, 2011. Как работает reCAPTCHA? URL: https://habr.com/ru/post/121010 (дата обращения 03.05.2020).
4. Habr, 2009. Тест Тьюринга. URL: https://habr.com/ru/post/69758 (дата обращения 01.05.2020).
5. Kaspersky, 2020. Что такое ботнет? URL: https://www.kaspersky.ru/resource-center/threats/botnet-attacks (дата обращения 01.05.2020).
6. Pycharm, 2020. Jetbrains Developer Tool. URL:https://www.jetbrains.com/pycharm (дата обращения 03.05.2020).
7. reCAPTCHA, 2020. The new way to stop bots. URL: https://www.google.com/recaptcha/intro/v3.html (дата обращения 03.05.2020).
8. Rucaptcha. 2020. URL: https://rucaptcha.com/software/category/skripti-i-biblioteki (дата обращения 03.05.2020).
9. Sololearn Python, 2020. URL: https://www.sololearn.com/Play/Python (дата обращения 03.05.2020).
10. Turing test, 2020. From Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Turing_test (дата обращения 03.05.2020).
11. Пудовикова, П.Д., Титов, С.С., 2016. Метод реализации captcha на основе подбора области изображения. URL: https://elibrary.ru/item.asp?id=28335822 (дата обращения 07.05.2020).

References

1. Anaconda, 2020. URL: https://www.anaconda.com/ (date of circulation: 05.05.2020).
2. CAPTCHA, 2020. Telling Humans and Computers Apart Automatically. URL: http://captcha.net (date of circulation: 03.05.2020).
3. Habr, 2011. How it works reCAPTCHA? URL: https://habr.com/ru/post/121010 (date of circulation: 03.05.2020).
4. Habr, 2009. Turing test. URL: https://habr.com/ru/post/69758 (date of circulation: 01.05.2020).
5. Kaspersky, 2020. What is a botnet? URL: https://www.kaspersky.ru/resource-center/threats/botnet-attacks (date of circulation: 01.05.2020).
6. Pycharm, 2020. Jetbrains Developer Tool. URL: https://www.jetbrains.com/pycharm (date of circulation: 03.05.2020).
7. reCAPTCHA, 2020. The new way to stop bots. URL: https://www.google.com/recaptcha/intro/v3.html (date of circulation: 03.05.2020).
8. Rucaptcha, 2020. URL: https://rucaptcha.com/software/category/skripti-i-biblioteki (date of circulation: 03.05.2020).
9. Sololearn Python, 2020. URL: https://www.sololearn.com/Play/Python (date of circulation: 03.05.2020).
10. Turing test, 2020. From Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Turing_test (date of circulation: 03.05.2020).
11. Pudovikova P.D., Titov S.S., 2016. Method of implementation captcha based selection of the images URL: https://elibrary.ru/item.asp?id=28335822 (date of circulation: 07.05.2020).

Девицына Светлана Николаевна, кандидат технических наук, доцент, доцент кафедры Информационная безопасность Института радиоэлектроники и информационной безопасности
Гоголь Андрей Сергеевич, студент 4 курса кафедры Информационная безопасность Института радиоэлектроники и информационной безопасности

Devitsyna Svetlana Nikolaevna, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department Information security, Institute of Radioelectronics and Information security
Gogol Andrey Sergeevich, 4th year student of the Department Information security, Institute of Radioelectronics and Information security