

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS - UNIFESO
CENTRO DE CIÊNCIA E TECNOLOGIA - CCT
CURSO DE BACHARELADO EM CIÊNCIA DA
COMPUTAÇÃO

TAYLANE BRANDÃO NEVES

ANÁLISE E APLICAÇÕES METODOLÓGICAS PARA A
QUEBRA DE CAPTCHAS

TERESÓPOLIS
2020

TAYLANE BRANDÃO NEVES

ANÁLISE E APLICAÇÕES METODOLÓGICAS PARA A
QUEBRA DE CAPTCHAS

**Trabalho de Conclusão de Curso sub-
metido ao Centro Universitário Serra
Dos Órgãos, como requisito necessário
para obtenção do grau de Bacharel em
Ciência da Computação**

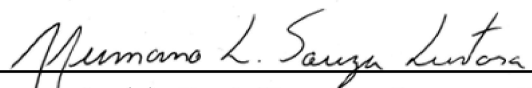
Teresópolis, novembro de 2020

N428	<p>Neves, Taylane Brandão. Análise e aplicações metodológicas para a quebra de captchas./Taylane Brandão Neves. – 2020. 42f.</p> <p>Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Serra dos Órgãos, UNIFESO, Teresópolis, 2020. Bibliografia: f. 41-42. Orientador: Prof. Hermano Lustosa.</p> <p>1-Ciências da Computação. 2. “CAPTCHAs”. 3. “GSA” 4. “reCAPTCHA”. 5. “Reconhecimento Óptico”. I. Título.</p> <p>CDD 004</p>
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CENTRO UNIVERSITÁRIO SERRA DOS ÓRGÃOS

TAYLANE BRANDÃO NEVES

Esta Monografia foi julgada adequada para a obtenção do título de Bacharel em Ciência da Computação, sendo aprovada em sua forma final pela banca examinadora:



Orientador(a): Prof. Hermano Lustosa, DSc
Centro Universitário Serra dos Órgãos -
UNIFESO



Prof. Laion Luiz Fachini Manfro
Instituto Militar de Engenharia, MSc - IME



Prof Dr. Alberto Angonese, DSc
Instituto Militar de Engenharia - IME

Teresópolis, 24 de novembro de 2020

Agradecimentos

Agradeço ao Prof. Hermano pela orientação neste trabalho e pelas melhores aulas de Sistemas Operacionais possíveis.

Aos Professores Laion e Alberto, que além de aceitarem fazerem parte da minha banca, foram modelos excepcionais tanto no âmbito acadêmico como no âmbito pessoal, além das inúmeras oportunidades profissionais que me foram concedidas.

A minha mãe Rejane Brandão a quem devo tudo que conquistei e pelo apoio ao meu sonho de me tornar programadora.

Resumo

Tendo em vista a quantidade crescente de dados a disposição na internet é seguro afirmar que também aumenta a necessidade de proteger os websites de acessos massivos realizados por robôs, para tal, utilizam-se diversas tecnologias, uma delas é o CAPTCHA (teste de Turing público completamente automatizado para diferenciação entre computadores e humanos). Atualmente é possível encontrar CAPTCHAs em diversos websites, desde serviços governamentais até comércios eletrônicos. Este trabalho tem como intuito realizar um estudo sobre CAPTCHAs, sobretudo naqueles que são baseados em texto. Demonstrando como é possível burlar esses dispositivos de segurança, explicando os métodos e realizando uma comparação entre eles.

Palavras-chave: CAPTCHAs, GSA, reCAPTCHA, Reconhecimento Óptico de Caracteres e Quebra de CAPTCHA.

Abstract

Considering the constant growing of data available online is possible to assume that the necessity of protecting websites from massive access made by robots is also growing, one of the ways of doing this are CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). Currently a large number of websites, from governmentals sites to e-commerces, use this device. This paper aims to study CAPTCHAs, focusing on text baseds; More specifically the goal is to solve thoses security mechanism, explaining methods to do so and comparing them to each other.

Keywords: CAPTCHAs, GSA, reCAPTCHA, Optical character recognition and CAPTCHA solving.

Lista de ilustrações

Figura 1 – Exemplo de um CAPTCHA textual.	14
Figura 2 – Exemplos de FunCAPTCHA.	15
Figura 3 – Exemplo de um MathCAPTCHA	16
Figura 4 – Exemplo de 15 CAPTCHAs textuais utilizados na internet.	16
Figura 5 – Exemplo de um reCAPTCHA V1.	18
Figura 6 – reCAPTCHA V1 e suas quatro variações mais comuns.	18
Figura 7 – Exemplo de um reCAPTCHA V2.	19
Figura 8 – Exemplo de um reCAPTCHA V2 após a checagem.	19
Figura 9 – Exemplo de um reCAPTCHA V2 expandido.	20
Figura 10 – Logo do reCAPTCHA v3.	20
Figura 11 – Tela inicial do GSA.	22
Figura 12 – Janela de configuração do emulador de <i>services</i> do GSA.	23
Figura 13 – Janela de edição de SDKs do GSA.	23
Figura 14 – Comparação entre os tempos de cinco <i>services</i>	27
Figura 15 – Comparação entre os custos de cinco <i>services</i>	27
Figura 16 – Interface do Insomnia com a requisição de quebra.	28
Figura 17 – Website Wikipédia apresentando um desafio de CAPTCHA.	29
Figura 18 – Exemplos de 5 CAPTCHAs do site da Wikipedia.	30
Figura 19 – Exemplo de CAPTCHA referente ao SDK Bestallsharp.	30
Figura 20 – Website AliExpress apresentando um desafio de CAPTCHA.	31
Figura 21 – Exemplos de 5 CAPTCHAs do site da AliExpress.	31
Figura 22 – Exemplo de CAPTCHA referente ao SDK HarveyDong.	32
Figura 23 – Exemplo do tratamento realizado para a obtenção de imagens de um reCAPTCHA V1.	33
Figura 24 – Site do Reclame Aqui e o respectivo código fonte ao lado.	34
Figura 25 – Gráfico de comparação de acurácia entre métodos de quebra para CAPTCHAs textuais.	36
Figura 26 – Gráfico de comparação de tempo entre métodos de quebra para CAPT- CHAs textuais.	37
Figura 27 – Gráfico de comparação de acurácia entre métodos de quebra.	38
Figura 28 – Gráfico de comparação de tempo entre métodos de quebra.	39

Lista de abreviaturas e siglas

CAPTCHA	Completely Automated Public Turing Tests to tell Computers and Humans Apart
API	Application Programming Interface
OCR	Optical Character Recognition
IA	Inteligência Artificial
SDK	Software development kit

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos e Organização	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Tipos de CAPTCHAs	14
2.1.1	Textual:	14
2.1.2	ReCAPTCHA:	15
2.1.3	FunCAPTCHA:	15
2.1.4	MathCAPTCHA:	15
2.2	CAPTCHAs Textuais	16
2.2.1	Corpo	16
2.2.2	Background	17
2.3	reCAPTCHA	17
2.3.1	Origem	17
2.3.2	Evolução	17
2.3.2.1	reCAPTCHA V1	17
2.3.2.2	reCAPTCHA V2	19
2.3.2.3	reCAPTCHA V3	20
2.4	Processos de quebra	21
2.4.1	<i>Services</i>	21
2.4.2	Softwares de quebra	21
2.4.3	<i>OCR</i>	24
2.4.4	Inteligência Artificial	24
3	METODOLOGIA	26
3.1	Quebra de CAPTCHAs textuais	29
3.1.1	CAPTCHA Wikipédia	29
3.1.2	CAPTCHA AliExpress	30
3.2	Quebra de reCAPTCHAs	32
3.2.1	v1	32
3.2.2	v2	33
3.2.3	v3	34
3.3	Quebra de FunCAPTCHAs	35
4	RESULTADOS	36
4.1	Resultados CAPTCHAs Textuais	36

4.2	Resultados reCAPTCHA	38
4.3	Resultados FunCAPTCHA	38
4.4	Resultados Gerais	38
5	CONCLUSÃO	41
	REFERÊNCIAS	42

1 INTRODUÇÃO

A proteção de dados e informações na internet sempre foi uma preocupação relevante de um modo geral, buscando-se sempre alternativas e meios de proteger tais dados, tentando assegurar totalmente ou dificultar o acesso a informações sensíveis [Morein et al. 2003]. Focando principalmente em evitar tráfegos de informações feitos por robôs e através de *scripts* automáticos de captação de dados, popularmente conhecidos como *Crawlers*. Entretanto, devido aos malefícios que tais atividades podem causar, gerando acessos massivos e velozes de forma a desestruturar as aplicações web ou *websites*, procuram-se sempre maneiras de evitar que tais robôs naveguem pela aplicação;

CAPTCHAs são aplicados justamente nesse intuito, de apresentar um desafio que consiga diferenciar se o tráfego está sendo realizado por humanos ou por robôs, sendo cada vez mais popularizados e aplicando técnicas de segurança cada vez melhores.

Sendo presentes até mesmo em sites governamentais brasileiros, tal qual a Receita Federal ou sites de abrangência municipal e estadual. Normalmente, protegendo áreas de consulta de CPFs e CNPJs ou consultas a notas fiscais.

O uso do CAPTCHA realmente funciona, ele diminui ou anula completamente o acesso de robôs aos sites [Morein et al. 2003]. Gerando apenas como efeito colateral um certo desconforto por parte dos usuários humanos que muitas vezes não compreendem por que um desafio está sendo proposto ou possuem seus fluxos de acesso atrapalhados.

Por outro lado, existem ocasiões em que é necessário burlar tais dispositivos de segurança, seja quando um usuário tem dificuldades de resolve-lo da forma usual, como pode ser visto no artigo [Yan e Ahmad 2008], ou quando se precisa automatizar um processo ou fluxo de acesso a dados. Com esta proposição, através da análise de CAPTCHAs e quebras para os mesmos, este trabalho demonstra quais os métodos mais eficazes para burlar cada tipo de CAPTCHA citado.

Durante este estudo, foi possível concluir a eficácia e flexibilidade dos serviços de quebra utilizando a força de trabalho humana. Além de notar como os CAPTCHAs desafiam a visão computacional, seja em *softwares* de quebra ou por meio da inteligência artificial.

1.1 Objetivos e Organização

Entre os objetivos do presente trabalho estão: apresentar CAPTCHAs de uma forma geral, realizar uma análise e descrição dos diferentes tipos de CAPTCHA, fazendo um levantamento das principais técnicas e ferramentas para quebra e uma avaliação

comparativa entre elas. Apresentando, de certa forma, os riscos em se utilizar apenas CAPTCHAs para a proteção de websites e quais CAPTCHAs possuem um maior nível de segurança. Além de mostrar quais os métodos de quebra são mais eficazes. Auxiliando o leitor, por exemplo, a escolher o dispositivo de segurança mais relevante a ser utilizado em uma aplicação de sua autoria.

O trabalho está organizado em cinco capítulos, sendo eles: Introdução, Fundamentação Teórica, Processos de Quebra, Metodologia, Resultados e Conclusão.

Abaixo uma relação dos capítulos: Introdução: Apresentação do contexto do trabalho e de sua problemática e descrição dos objetivos. Fundamentação Teórica: Descrição dos tipos de CAPTCHA mais amplamente utilizados na internet, o detalhamento de CAPTCHAs textuais, reCAPTCHAs e FunCaptchas, além de um breve detalhamento sobre os métodos de quebra. Metodologia: Apresentação dos critérios avaliados para o desempenho, uma breve análise entre os *services* de quebra, descrição dos testes de quebra realizados e a demonstração do processo. Resultados: Análise dos resultados obtidos com a metodologia utilizada. Conclusão: Apresentação das conclusões obtidas embasadas nos resultados alcançados.

2 FUNDAMENTAÇÃO TEÓRICA

Um CAPTCHA consiste em um dispositivo de segurança para a aferir se a entidade que está interagindo com o computador naquele determinado momento é um ser humano ou uma máquina [Ahn, Blum e Langford 2004]. Seu significado origina-se de "teste de Turing público completamente automatizado para diferenciação entre computadores e humanos". CAPTCHAs são principalmente aplicados na segurança web para bloquear ações automatizadas, como por exemplo, limitar o envio de e-mails maliciosos, compras fraudulentas em lojas online, evitar criação de contas falsas e postagens repetitivas em redes sociais ou até mesmo evitar trapaças em jogos eletrônicos.

O processo de segurança de um CAPTCHA é feito através de um desafio cognitivo, podendo variar entre visual e auditivo, dependendo do tipo de CAPTCHA, variando até mesmo o nível de dificuldade e segurança do mesmo.

O surgimento, dos CAPTCHAs "rudimentares" deu-se na década de 80 ao tentar camuflar o texto real utilizando caracteres que formavam o texto apenas visualmente, conhecido como "*leetspeak*", surgiu tanto como uma linguagem popular apenas para os usuários da internet quanto como um desafio de leitura e reconhecimento para máquinas [Ferrante e Ferrante 2008]. Entretanto como tal técnica poderia ser facilmente burlada, desde então diversos tipos de desafios foram sendo criados e refinados, alguns deles serão melhor descritos na Seção 2.1.

2.1 Tipos de CAPTCHAs

2.1.1 Textual:

O tipo de CAPTCHA mais comumente utilizado pela internet desde sua concepção [Chow e Susilo 2017].

Figura 1 – Exemplo de um CAPTCHA textual.



(Fonte: <https://upload.wikimedia.org/wikipedia/commons/6/69/Captcha.jpg>, acesso em 23/06/2020.)

Conforme demonstrado na Figura 1, um CAPTCHA textual apresenta uma imagem com um texto, normalmente com ruídos ou outro tipo de técnica que dificulte o reconhecimento computacional. Possuindo inúmeras variações, cada um deles explorando de uma forma diferente as inúmeras limitações das OCRs (Reconhecimento óptico de caracteres), também conhecido como visão computacional.

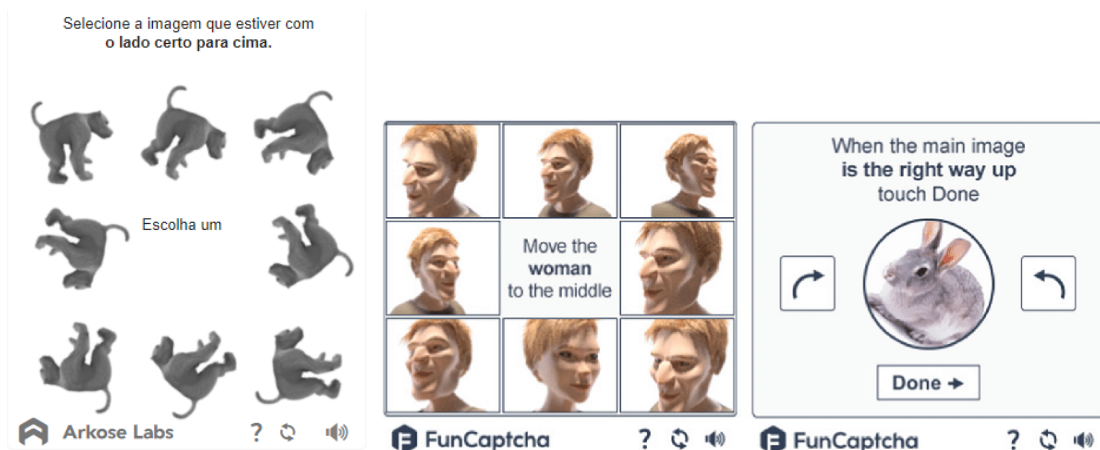
Será o principal abordado neste trabalho, sendo descrito na Seção 2.2 e tendo sua quebra demonstrada na Seção 3.1.

2.1.2 ReCAPTCHA:

Possuindo 3 versões, sendo uma delas textual e duas outras de reconhecimento de imagem. Serão detalhadas mais a frente na Seção 2.3.

2.1.3 FunCAPTCHA:

Figura 2 – Exemplos de FunCAPTCHA.



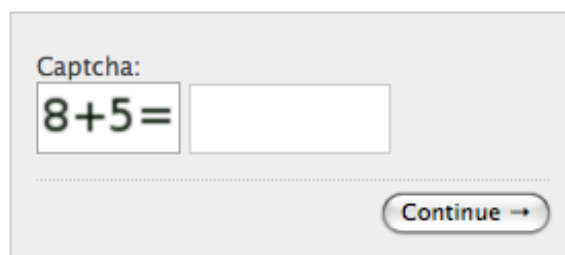
(Fonte: <https://funcaptcha.com/fc/api/nojs>, acesso em 28/07/2020.)

Possui uma imagem que deve ser rotacionada, como visto na Figura 2, até que esteja alinhada horizontalmente, possui o intuito de ser divertido, além de aumentar a dificuldade pois além de reconhecer a imagem, o computador ainda precisa ser capaz de alinhá-la.

2.1.4 MathCAPTCHA:

Consiste em um problema matemático simples, como o da Figura 3, que deve ser resolvido pelo usuário, sendo apenas desafiante para o reconhecimento de imagem e não para um ser humano.

Figura 3 – Exemplo de um MathCAPTCHA

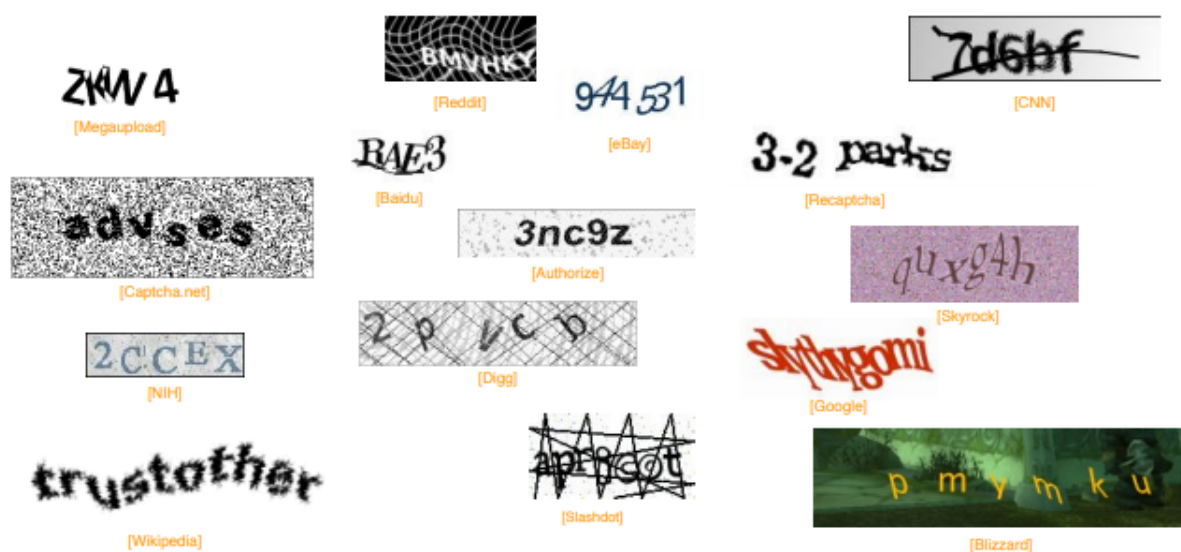


(Fonte: <https://developers.google.com/recaptcha>, acesso em 28/07/2020.)

2.2 CAPTCHAs Textuais

Existem diversas variações desta aplicação de CAPTCHA, conforme demonstrado na Figura 4, podendo ser chamado de esquema ou tipo. Pode-se dividir um CAPTCHA textual em fundo (background) e corpo:

Figura 4 – Exemplo de 15 CAPTCHAs textuais utilizados na internet.



(Fonte: Próprio autor.)

2.2.1 Corpo

No corpo estão presentes os caracteres que devem ser reconhecidos, podendo variar em quantidade e tipos. Por exemplo, como a Figura 4 demonstra, uma variação de CAPTCHA pode conter quatro caracteres letras e números enquanto outro possuir seis caracteres e apenas letras. Tais caracteres podem possuir ruídos, tal qual riscos, ondulações, estarem aglomerados ou rotacionados, como forma de dificultar a leitura das OCRs, o que acaba dificultando a leitura dos seres humanos por consequência.

2.2.2 Background

O plano de fundo dos CAPTCHAs também influencia muito o reconhecimento, por exemplo, um fundo colorido ou com padrões dificulta o reconhecimento da OCR em o que é caractere e o que é ruído.

CAPTCHAs monocromáticos (caracteres pretos com o fundo branco), por exemplo, acabam interferindo em uma das principais técnicas de processamento de imagens para a quebra, a limiarização ou binarização da imagem, deste modo cada pixel ou será completamente branco ou completamente preto. O objetivo deste tratamento é destacar os caracteres das sujeiras da imagem [Gao et al. 2013].

2.3 reCAPTCHA

2.3.1 Origem

Originalmente os reCAPTCHAs foram criados para possibilitar a digitalização de livros, até então irreconhecíveis por escâneres de computadores, através de um esquema de contribuição colaborativa em que imagens representando palavras fossem apresentadas aos usuários, fazendo assim que de palavra em palavra fosse possível digitalizar todo um livro. O reCAPTCHA foi criado ao fim de 2009 pelos desenvolvedores: Luis von Ahn, David Abraham, Manuel Blum, Michael Crawford, Ben Maurer, Colin McMillen e Edison Tan da Universidade Carnegie Mellon nos Estados Unidos Da América [Sivakorn, Polakis e Keromytis 2016].

Porém o propósito original de criação dos reCAPTCHAs se diferem um pouco do uso atual, em 2014 o projeto foi adquirido pelo Google, responsável pelo mesmo até o presente momento, para ser utilizado no "*Google Books*" recebendo diversas modificações e melhoramentos, tal qual as variações intituladas v2 e v3, que serão detalhadas no próximo tópico. Até o ano de 2018, onde o projeto do v1 foi oficialmente encerrado, foram exibidos mais de 100 milhões de CAPTCHAs diariamente, dessa forma foi possível digitalizar diversos livros.

2.3.2 Evolução

2.3.2.1 reCAPTCHA V1

Em seu formato original eram apresentadas ao usuário duas imagens em texto, conforme demonstra a Figura 5, a primeira imagem possuía um texto que já havia sido reconhecido previamente, sendo utilizado como método de controle de erros e a segunda imagem sendo o texto passível de reconhecimento durante aquela iteração.

Figura 5 – Exemplo de um reCAPTCHA V1.



(Fonte: <https://support.google.com/recaptcha/?hl=en>, acesso em 23/06/2020.)

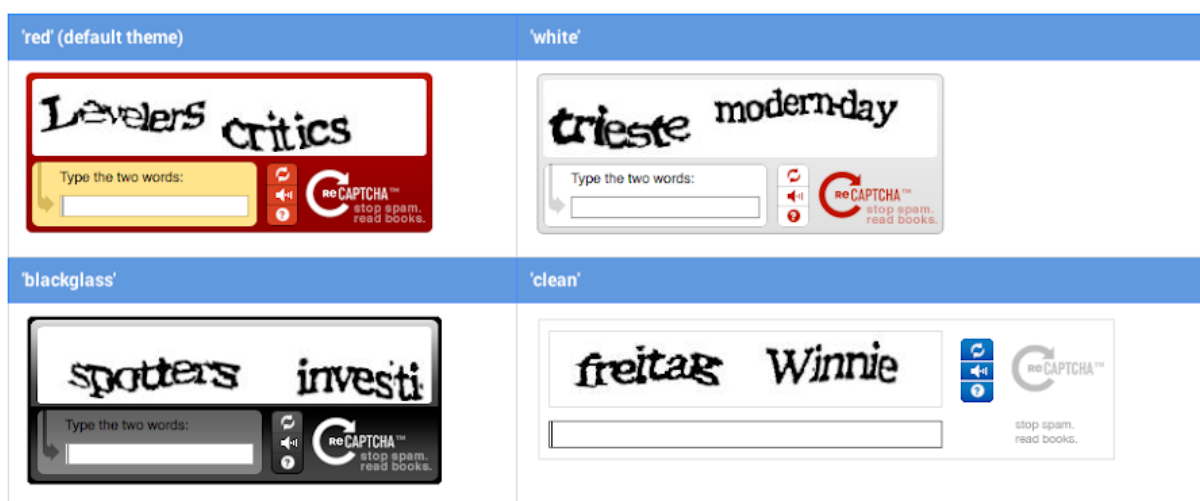
O usuário então deve inserir na caixa de texto os caracteres correspondentes aos que aparecem na imagem. Ao lado da caixa são apresentados três botões em alinhamento vertical:

- Obter um novo desafio;
- Ouvir o desafio em forma de áudio;
- Obter informações sobre *reCAPTCHA*s;

E no canto extremo direito é apresentado o logo do projeto juntamente da frase "*Stop spam, read books.*" em tradução livre "Acabe com o *spam*, leia livros", evocando ao leitor o objetivo do desafio sendo apresentado, como impedindo atividade irregular nos *websites* e ao mesmo tempo inspirando o usuário a leitura pois o mesmo estava ajudando no projeto de digitalização de um livro, mesmo que sua participação fosse involuntária.

O mesmo também possuía 4 temas de colorações, vide Figura 6, possibilitando uma maior acomodação sensorial ao ser inserido no website.

Figura 6 – reCAPTCHA V1 e suas quatro variações mais comuns.

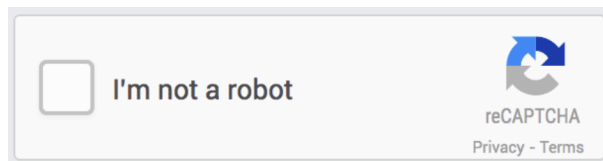


(Fonte: <https://support.google.com/recaptcha/?hl=en>, acesso em 20/09/2020.)

Posteriormente, em 2012, começaram a ser exibidas fotografias tiradas pelo projeto do Google Street View, apresentando placas e números de casas, ajudando no reconhecimento de endereços. Sendo oficialmente encerrado em 31 de março de 2018, quando o V2 e V3 já existiam e estavam sendo amplamente utilizados.

2.3.2.2 reCAPTCHA V2

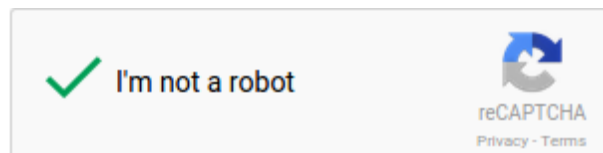
Figura 7 – Exemplo de um reCAPTCHA V2.



(Fonte: <https://support.google.com/recaptcha/?hl=en>, acesso em 20/09/2020.)

A segunda versão do sistema consistia em apresentar uma caixa de seleção ao usuário, como na Figura 7, e o mesmo precisaria apenas clicar nesta caixa e esperar que fosse validado ou não, tal validação é feita baseada em métricas do que o mecanismo considera como uma suposta atividade de *bot*.

Figura 8 – Exemplo de um reCAPTCHA V2 após a checagem.



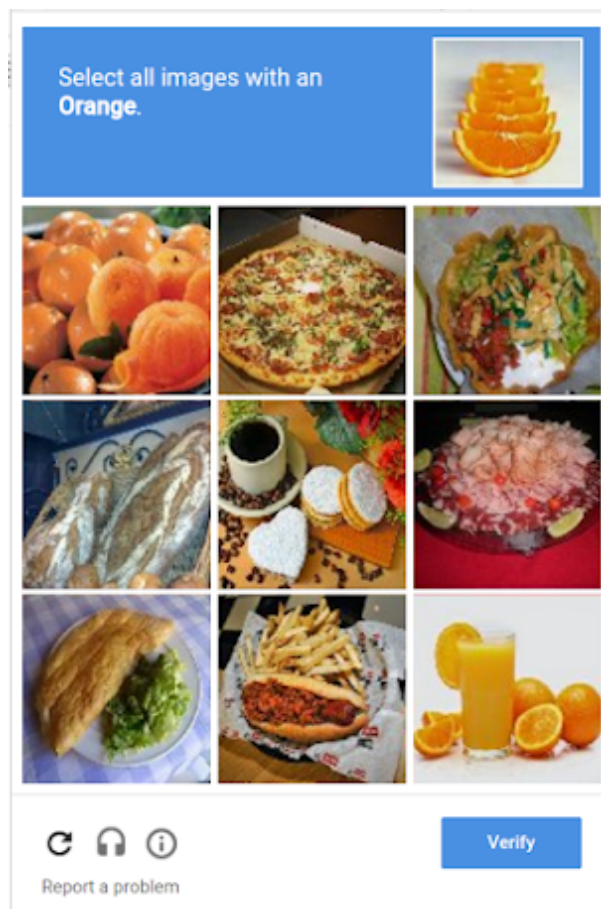
(Fonte: <https://support.google.com/recaptcha>, acesso em 20/09/2020.)

Caso a entidade resolvendo o desafio fosse considerado um ser humano, seria apresentada uma verificação como mostra a Figura 8 e o mesmo poderia prosseguir com a utilização do *website*, caso sua atividade seja considerada suspeita será apresentado outro desafio, sendo este, uma tela para reconhecimento de imagens como a Figura 9 demonstra, tendo como título uma frase com uma exemplificação de imagem que deveria ser correspondida com outras nove imagens exibidas em grade. O desafio consiste em que o usuário seja capaz de clicar nas imagens similares a primeira apresentada.

Em 2007 o Google lançou o reCAPTCHA invisível, sua invisibilidade se devia a não ser necessário marcar a caixa de seleção para prosseguir o fluxo de navegação, o mesmo só seria exibido caso o usuário fosse considerado de alto risco.

Como os dados coletados através do reCAPTCHA são amplamente utilizados para o treinamento da inteligência artificial do Google, o conjunto de imagens apresentados inicialmente consistia em itens do cotidiano, tal qual gatos, café, chá e flores. Com a evolução do poder de reconhecimento de imagens itens mais complexos com contexto voltado ao trânsito começaram a ser apresentados, tal qual carros, ônibus e semáforos.

Figura 9 – Exemplo de um reCAPTCHA V2 expandido.



(Fonte: <https://support.google.com/recaptcha>, acesso em 20/09/2020.)

Desta forma esse reconhecimento ajuda no projeto de treinamento de carros autônomos do Google.

2.3.2.3 reCAPTCHA V3

Figura 10 – Logo do reCAPTCHA v3.



(Fonte: <https://support.google.com/recaptcha>, acesso em 20/09/2020.)

A versão do reCAPTCHA chamada de v3, com a logo demonstrada na Figura 10, foi lançada em outubro de 2018, o mesmo não utiliza desafios a serem apresentados ao usuário nem mesmo possui uma representação visual, sendo um serviço a ser implementado no website onde serão captados informações referentes a navegação do usuário, desta forma, o sistema classifica o usuário com uma nota, cabendo ao administrador do site decidir

o que fazer com cada classificação que o sistema fará. Podendo assim ser mais flexível, podendo-se aplicar níveis mais elevados de segurança a conteúdos mais sensíveis e níveis menos de segurança em conteúdos mais banais e corriqueiros. Dessa forma, um desafio tal qual o desafio V2 é exibido apenas se o usuário for detectado como suspeito.

2.4 Processos de quebra

Existem diversas formas de se quebrar um CAPTCHA, neste trabalho serão abordados os seguintes meios: Softwares, Services, OCR e inteligência artificial.

2.4.1 Services

Serviços de quebra de CAPTCHA fornecem uma forma de passar por diversos tipos de desafios, sejam eles CAPTCHAs textuais, reCAPTCHAs, FunCAPTCHAs ou outros. Existem dois tipos de *services*, os que funcionam com a força de trabalho humano e os que fornecem um tipo de computação em nuvem realizando a quebra por OCRs. Em ambos os casos, os *services* proveem *APIs* (Interface de Programação de Aplicativos). para que se possa realizar requisições, cada serviço fornece um fluxo próprio, como pode ser visto na documentação do *service* AntiCaptcha [Anticaptcha 2020]. Entretanto, será descrito funcionamento geral da quebra utilizando a força de trabalho humano, que é dado da seguinte forma:

- Cria-se uma tarefa (*Task*), especificando o tipo da mesma;
- Obtém-se um *ID* como retorno da criação;
- Realiza-se requisições de checagem para a *API* até que a resposta do desafio seja respondida, também conhecido como *Pooling*;
- A *API* irá notificar um humano que há um desafio disponível;
- O humano irá resolver o desafio e devolver a resposta à *API*;
- Com a resposta obtida da *API* é possível ultrapassar o desafio e continuar o fluxo de navegação;

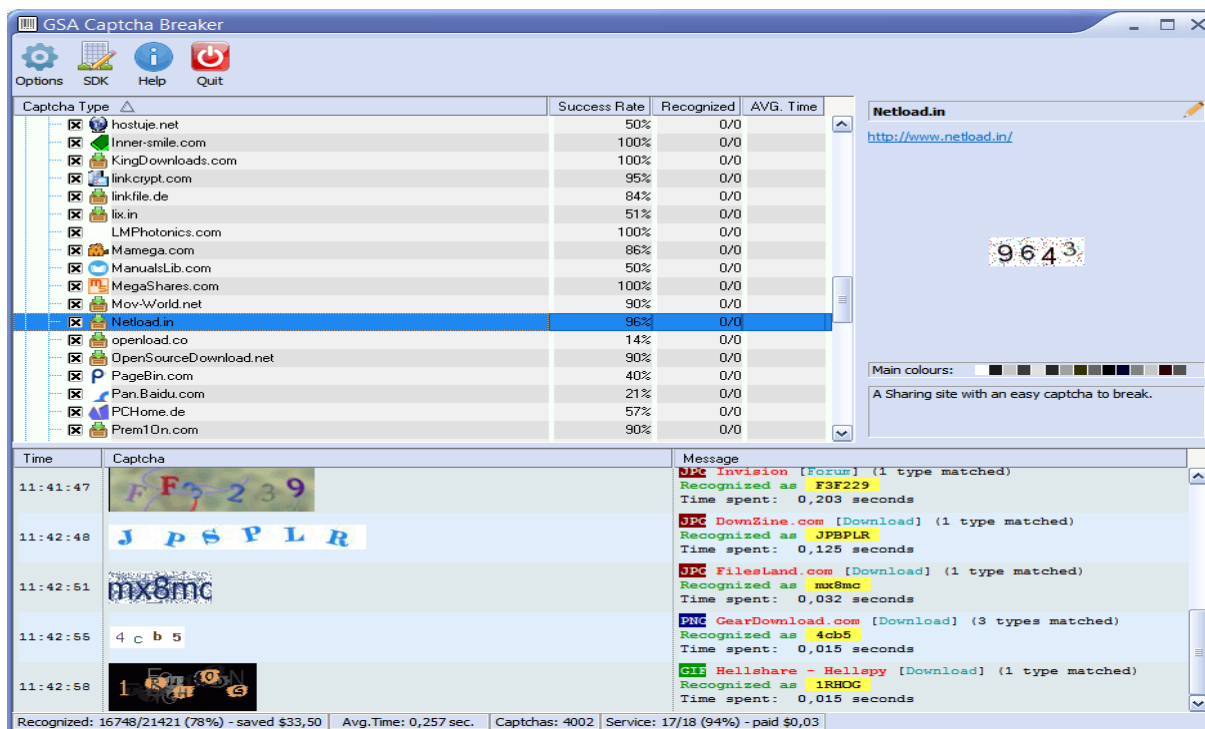
2.4.2 Softwares de quebra

Existem diversos softwares de quebra de CAPTCHA textuais disponíveis na internet para uso, sendo em sua maioria pequenos softwares feitos por indivíduos com intuito acadêmico ou experimental, sendo poucas as opções de softwares comerciais disponíveis para compra. Entre eles temos como mais relevantes o CAPTCHA Sniper [Captcha Sniper

2020] e o GSA Captcha Breaker [Sven Bansemer 2020], por possuírem comportamento semelhante, o segundo foi escolhido para ser detalhado neste trabalho.

Este software possui configurações de quebra chamadas de SDKs, sendo normalmente um SDK por tipo de CAPTCHA. Ao realizar o download do programa, o mesmo já vem com inúmeros SDKs configurações, sendo realizado atualizações periódicas adicionando mais itens.

Figura 11 – Tela inicial do GSA.

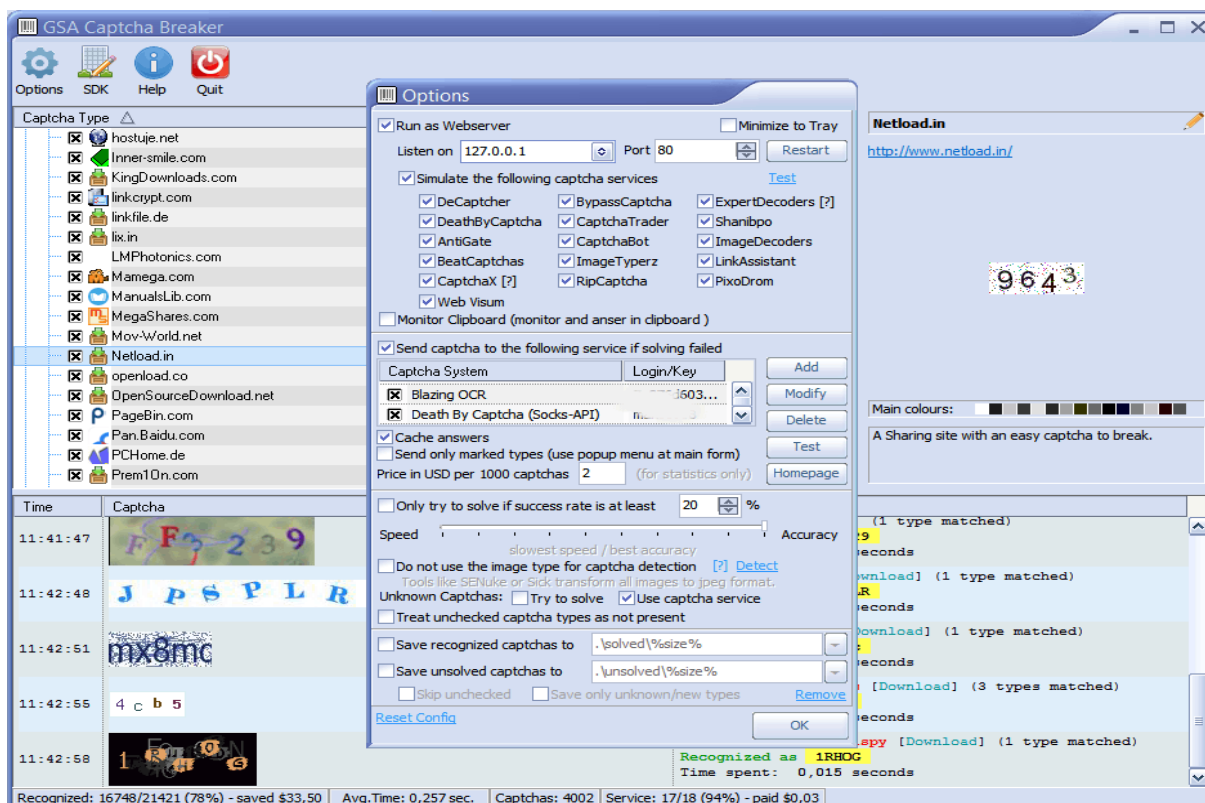


(Fonte: https://www.gsa-online.de/product/captcha_breaker, acesso em 15/10/2020.)

Na Figura 11 é possível visualizar a tela inicial do software. O menu da aplicação fica no canto superior esquerdo da imagem. Na caixa central são apresentados todos os SDKs previamente treinados tanto pelo usuário quanto os pré-definidos da aplicação. Na caixa inferior da tela são apresentados os CAPTCHAs que estão sendo quebrados pela aplicação em tempo real.

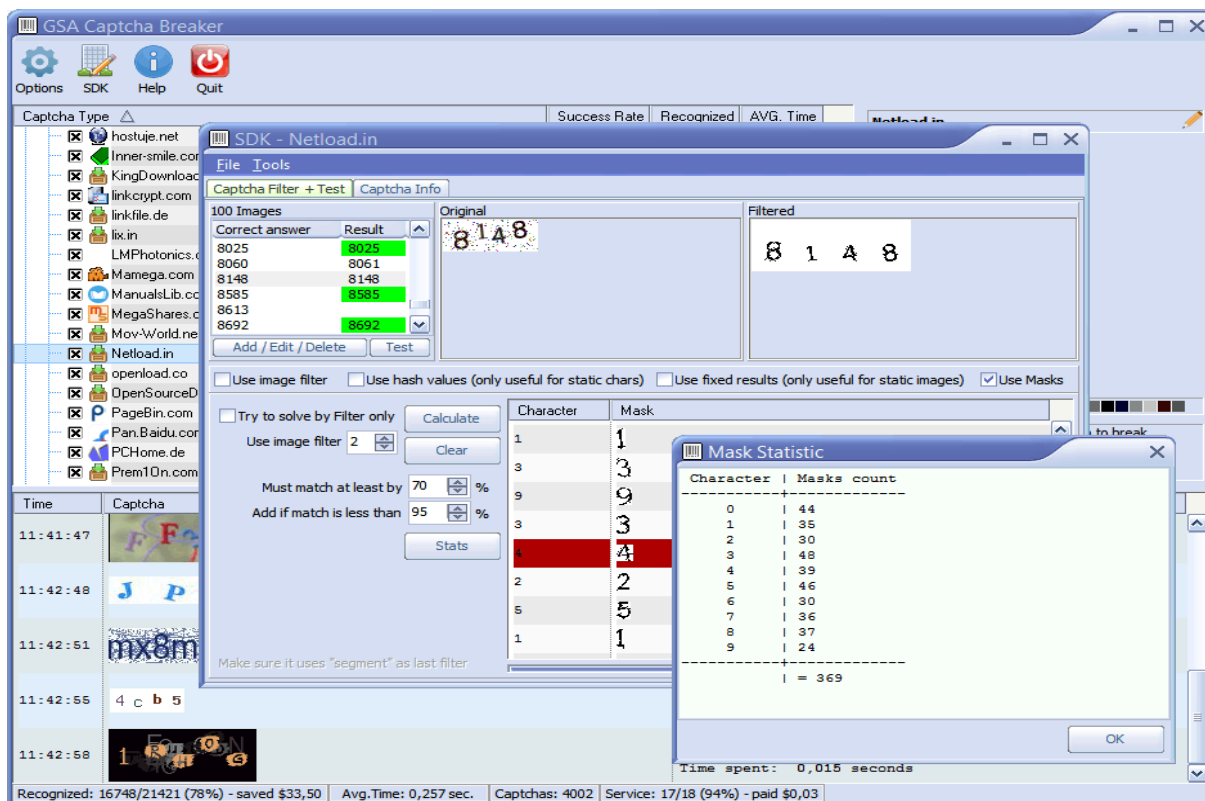
A aplicação possui a opção de simular uma API de quebra, tal qual um *service* pago, utilizando o mesmo protocolo do serviço simulado escolhido. Na Figura 12 é apresentado o menu de configuração do simulador de *services* sendo possível escolher mais de um ao mesmo tempo, de tal forma a aplicação responde a requisições no padrão de qualquer *service* selecionado. Nesta mesma tela também são apresentadas outras opções de como o software deve responder durante o processo de quebra.

A janela de edição de SDK permite o usuário criar um novo SDK ou editar um já existente. Conforme a Figura 13 demonstra.

Figura 12 – Janela de configuração do emulador de *services* do GSA.

(Fonte: https://www.gsa-online.de/product/captcha_breaker, acesso em 15/10/2020.)

Figura 13 – Janela de edição de SDKs do GSA.



(Fonte: https://www.gsa-online.de/product/captcha_breaker, acesso em 15/10/2020.)

2.4.3 OCR

O reconhecimento óptico de caracteres tem como intuito a conversão de texto visual, como fotos, documentos escaneados ou imagens de caracteres, em texto digital que pode ser editado, iterado para buscas ou armazenado de forma compacta [Arica e Vural 2012, p. 1]. Podendo ser aplicado de diversas formas, tal qual: Reconhecimento automatizado de placas de carros, possibilitar a pesquisa em textos impressos (Google Books) ou até mesmo como tecnologia assistiva para deficientes visuais.

O reconhecimento usa principalmente as duas técnicas a seguir:

- Por reconhecimento de padrões, onde os caracteres da imagem são isolados e realiza-se uma comparação do mesmo com fontes conhecidas, funcionando melhor com texto digitado e tendo pouca eficácia em textos escritos manualmente [Arica e Vural 2012, p. 2];
- Por reconhecimento de características, os caracteres são decompostos em partes menores, tal qual, linhas, intercessões e arredondamentos, criando uma lista de elementos que podem ser comparados com características conhecidas dos caracteres. Sendo eficiente também em textos escritos manualmente [Arica e Vural 2012, p. 2];

2.4.4 Inteligência Artificial

Uma das abordagens para a quebra de CAPTCHA, é o aprendizado de máquina, especificamente, utilizando o aprendizado profundo. Tal problema pode ser abordado focando na quebra de um tipo de CAPTCHA específico ou na quebra generalizada para diversos tipos.

No geral é necessário uma amostra considerável de variações de um mesmo tipo de CAPTCHA para se realizar o treinamento da inteligência artificial, ou seja, para se quebrar um CAPTCHA textual é necessário não apenas o treinamento para tal mas um treinamento específico para cada variação. O mesmo acontece para reCAPCHAs e outros CAPCHAs visuais, além do treinamento para reconhecimento de imagem é preciso utilizar uma base de dados de imagens similares a apresentadas pelo desafio. Ambos serão abordados na Seção 3.

Diversos artigos utilizam aprendizado profundo para a quebra de CAPCHAs, sendo o mais eficaz, como pode ser visto nos trabalhos "Deep-CAPTCHA: a deep learning based CAPTCHA solver for vulnerability assessment" [Noury e Rezaei 2020], "CAPTCHA Breaking with Deep Learning" [Zhao, Liu e Jiang 2017] e "CAPTCHA-22: Breaking Text-Based CAPCHAs with Machine Learning" [Labs 2019]. Ademais, o processo mais comum para o treinamento de uma rede neural para quebra necessita que o CAPTCHA seja

previamente conhecido e possua exemplares já resolvidos, a simplificação desse processo é dado da seguinte forma:

Utilizando o OpenCV, a biblioteca de código aberto de visão computacional (Open Source Computer Vision Library) [OpenCV 2020], tal biblioteca é utilizada para interagir com as imagens dos CAPTCHAs a serem quebrados. Ela fornece diversos utilitários que podem ser utilizados durante o processamento de imagem, como por exemplo: Reconhecer contornos, extremamente útil para reconhecer quais *pixels* pertencem ao um caractere (sem reconhecer o caractere em si). Ferramentas de cores, que podem ser utilizadas para mudar a coloração dos CAPTCHAs para preto e branco, facilitando o reconhecimento. Divisor ou concatenador de imagens, uteis para criar imagens individuais dos caracteres, sendo mais fácil para a OCR reconhece-los individualmente.

Comumente utiliza-se um arcabouço (*Framework*), ou seja, um conjunto de ferramentas que permite o treinamento de uma rede de aprendizado profundo. Entre eles, podem ser citados como exemplo: TensorFlow [TensorFlow 2020], TORCH/PyTorch [PyTorch 2020] e KERAS [Keras 2020].

A seguir, obtém-se um grande volume de CAPTCHAs, idealmente mais de mil exemplares já resolvidos, para que se possa extrair os caracteres em imagens individuais, criando assim, um banco de dados referente a cada carácter que poderá ser utilizado como referencia de comparação para a OCR. Existe uma relação direta entre a quantidade de CAPTCHAs utilizados e o aprimoramento do resultado, como pode ser visto no artigo [Salvatore 2017], utilizando mil exemplares, obteve-se 90% de resultados corretos, enquanto ao utilizar cinquenta mil exemplares essa porcentagem subiu para 95,1%.

Como passo final utiliza-se um grupo "teste" de CAPTCHAs também previamente quebrados, comparando os resultados obtidos pela rede com os resultados corretos de forma a acorrer o aperfeiçoamento da rede.

3 METODOLOGIA

A análise comparativa entre os métodos de quebra apresentada, utiliza-se de três alternativas: via *Software*, via API e via inteligência artificial. Sendo a análise de inteligência artificial baseada no trabalho correlato [Ye et al. 2018]. Serão demonstradas as quebras para todos os tipos de CAPTCHAs descritos na Seção 2.1, exceto o Math CAPTCHA. O presente capítulo está dividido em três partes. A primeira visa a quebra de CAPTCHAs textuais, a segunda parte analisa a quebra de reCAPTCHAs e a terceira e última demonstra a quebra de FunCAPTCHAs.

Serão medidos os desempenhos dos métodos de quebra em três quesitos:

- Tempo, quantos segundos foram gastos para realizar a quebra de cada CAPTCHA individualmente.
- Acurácia, dado quantidade de CAPTCHAs submetidos, qual a porcentagem deles possuíram as respostas corretas.
- Apenas para *services*, será declarado o custo em dólares.

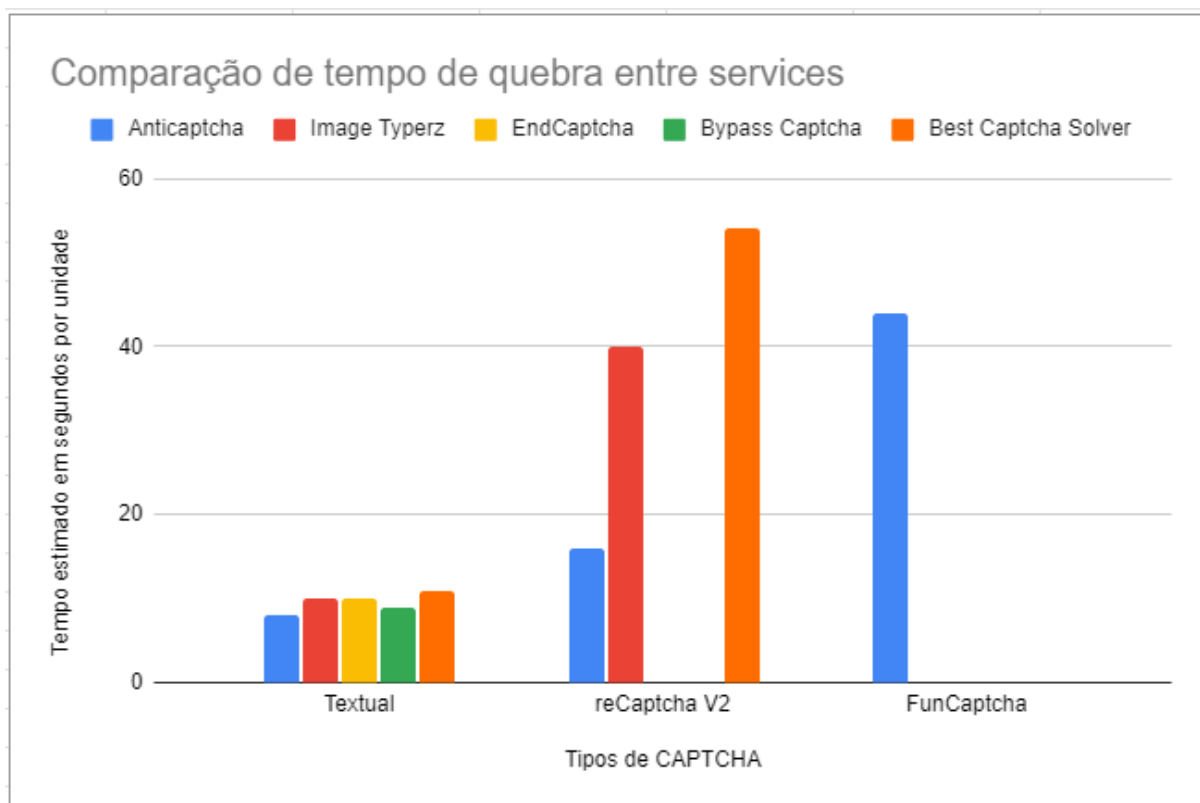
A quebra via API foi feita conforme está descrito na Seção 2.4.1, utilizando um *service* escolhido. A escolha foi motivada com base no tempo de resposta e custo oferecidos pelo *service*. Para exemplificação serão demonstrados dois gráficos referentes ao tempo e custo de cinco *services* disponíveis na internet. Todos os *services* escolhidos utilizam força de trabalho humana. Não foi medida a acurácia pois a maioria deles não declara tal dado em seus websites. Os demais dados não presentes no website foram omitidos.

Os cinco *services* escolhidos foram

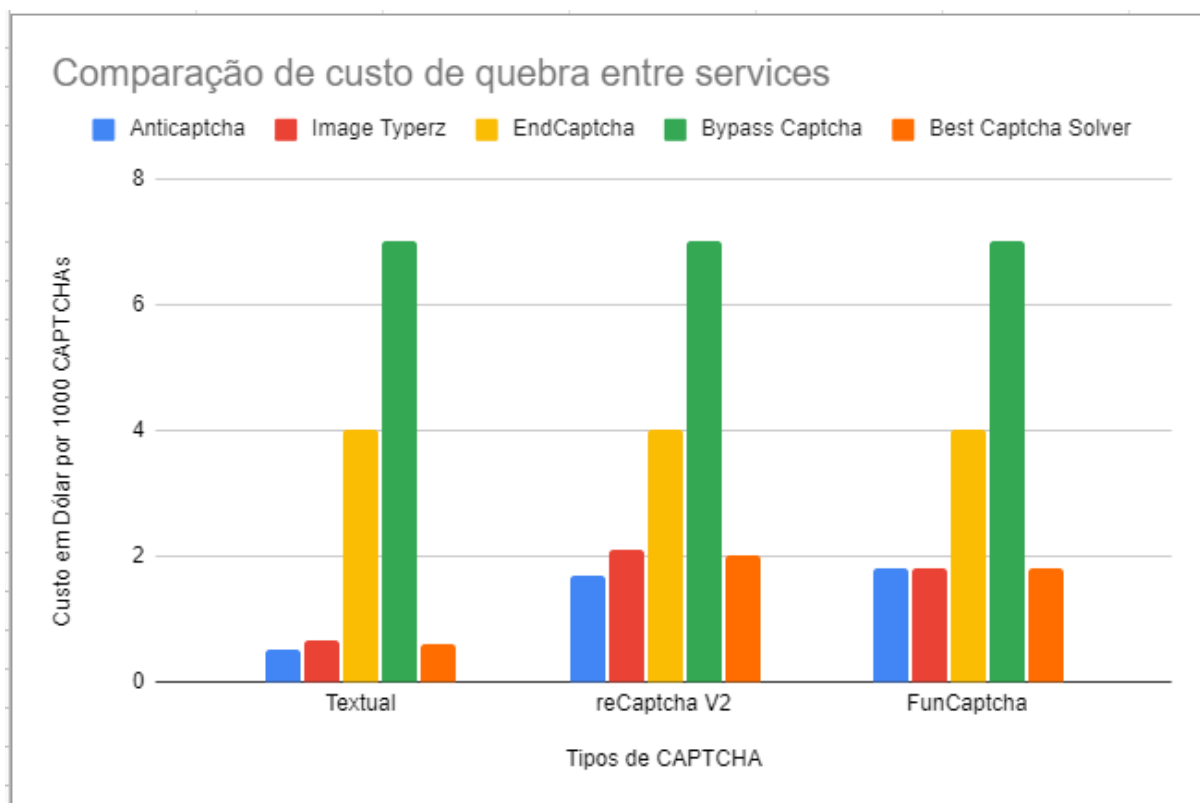
- Anticaptcha [Anticaptcha 2020];
- Image Typerz [Typerz 2020];
- EndCaptcha [EndCaptcha 2020];
- Bypass Captcha [2Captcha 2020];
- Best Captcha Solver [Solver 2020];

Todos os dados foram obtidos dos respectivos sites. Além disso, todas as informações são variantes e a maioria é modificada em tempo real

Ao fim do processo o *service* Anticaptcha foi escolhido por possuir o menor tempo, como demonstra a Figura 14. E possuir o menor custo (Figura 15).

Figura 14 – Comparação entre os tempos de cinco *services*.

(Fonte: Próprio autor.)

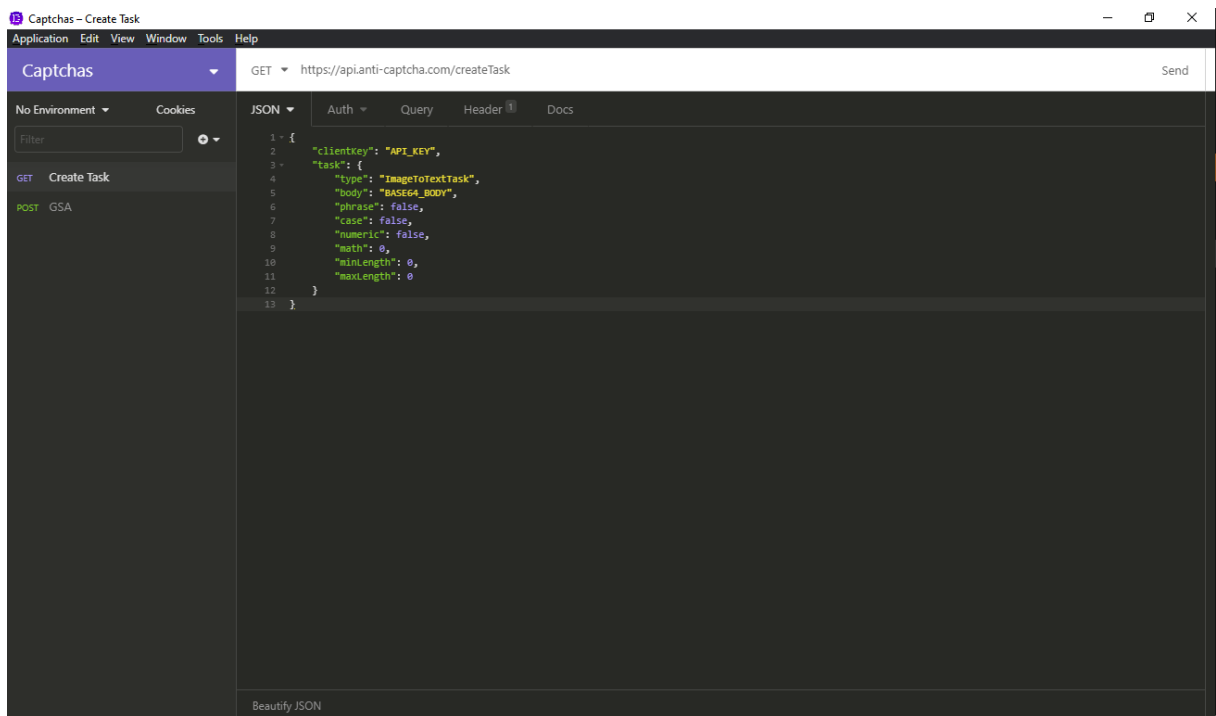
Figura 15 – Comparação entre os custos de cinco *services*.

(Fonte: Próprio autor.)

Além disso, em seu website o *service* alega ter um longo histórico de funcionamento, possuindo 99,9% de tempo no ar desde o ano de 2007 [Anticaptcha 2020], além de possuir um dos menores preços entre services similares.

Todas as requisições feitas ao serviço foram através do Insomnia, um *software* que fornece uma interface visual para a realização de requisições HTTP [Sebastian McKenzie 2020]. Para realizar a comunicação com o serviço, é necessário a criação de uma conta e adicionar créditos a mesma. Ao fim do processo, é gerada uma chave (API KEY), a qual deve ser enviada nas requisições de quebra. A Figura 16 demonstra a utilização da chave de quebra juntamente com a interface do software utilizado. O tempo e valor variam de acordo com o volume de dados processado pelo serviço naquele momento;

Figura 16 – Interface do Insomnia com a requisição de quebra.



(Fonte: Próprio autor.)

O procedimento da quebra via inteligência artificial é feito com base nos resultados do artigo [Ye et al. 2018]. Tal estudo foi feito por um grupo de oito indivíduos que prototiparam seu projeto apoiando-se em estudos correlatos e modelos matemáticos. Além disso, foi utilizado um número massivo de dados para a elaboração. Utilizando CAPTCHAs reais e gerados sinteticamente, ou seja, criados pelos próprios autores com o intuito de treinamento apenas. Utilizando um total de onze tipos (também chamado de esquema pelos autores) de CAPTCHAs populares na época. A quebra foi feita criando uma rede neural com duas camadas, a primeira chamada de resolvidor base e a segunda de resolvidor refinado. Para a primeira camada foram necessários duzentos mil CAPTCHAs sintéticos. Na segunda camada foram utilizados cerca de quinhentos CAPTCHAs de cada esquema para refinar a rede e mil CAPTCHAs para validar os testes. De tal forma, obtiveram uma

média de resolução de 36% com o resolvidor base e uma média de 61% com os 11 tipos de CAPTCHAs textuais testados;

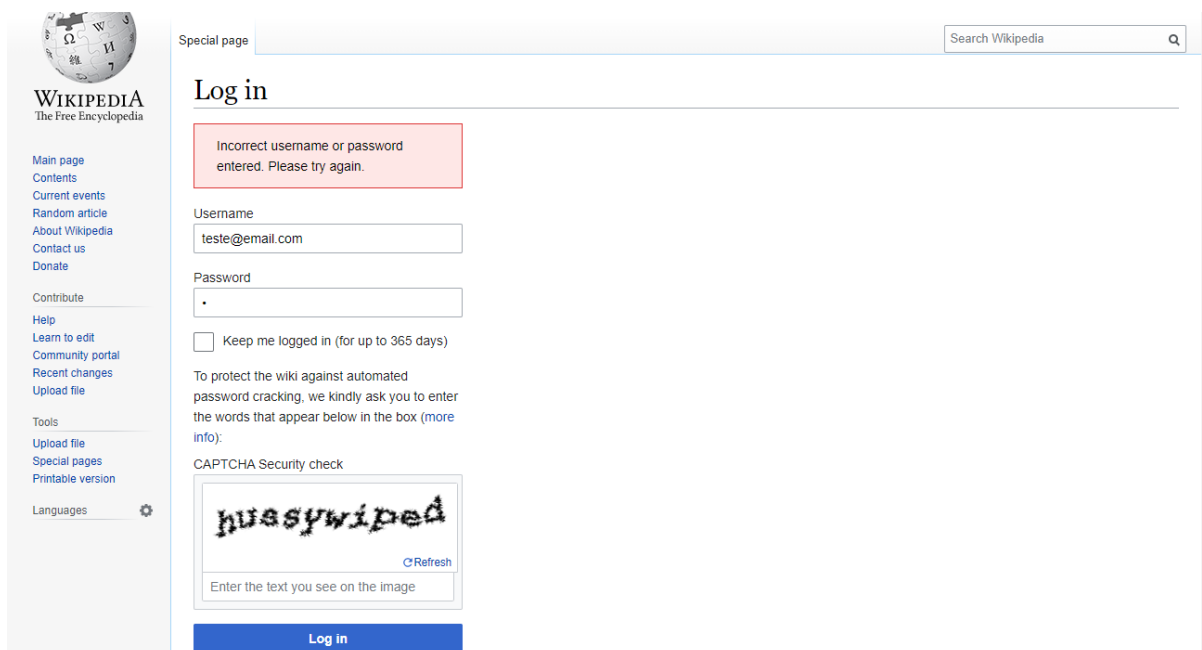
3.1 Quebra de CAPTCHAs textuais

A resolução de ambos os desafios textuais serão demonstradas utilizando os três métodos descritos na Seção anterior.

3.1.1 CAPTCHA Wikipédia

O primeiro CAPTCHA foi retirado do site da Wikipédia, uma enciclopédia online gratuita e colaborativa. Atualmente o site possui mais de 1,5 bilhões de acessos mensais [Wikipedia 2020], sendo utilizado a nível global. O CAPTCHA é apresentado na tela de login, conforme imagem 17, protegendo também a tela de criação de conta, no intuito de impedir a criação de usuários falsos, já que um usuário logado pode realizar alterações e contribuir para artigos do site. O CAPTCHA é de idealização e uso próprio.

Figura 17 – Website Wikipédia apresentando um desafio de CAPTCHA.

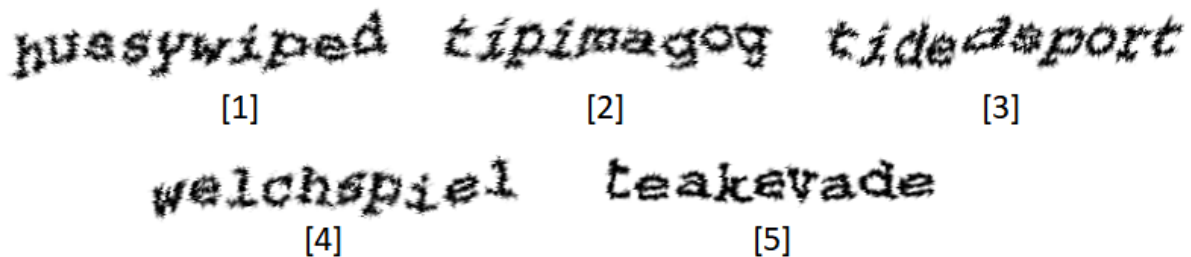


(Fonte: <https://en.wikipedia.org/w/index.php?title=Special:UserLogin&returnto=Main+Page>, acesso em 28/09/2020.)

Foram selecionados cinquenta amostras de CAPTCHAs para a realização da análise, a Figura 18 exemplifica cinco destes, cada um possui um tamanho diferente, variando de 236 a 274 para a largura e de 56 a 71 para a altura. Suas características de segurança são: a aglomeração de caracteres, dificultando a segmentação e reconhecimento individual dos

mesmos. A ondulação, distorção e um leve rotacionamento para dificultar o reconhecimento dos contornos do caractere. Possui letras apenas.

Figura 18 – Exemplos de 5 CAPTCHAs do site da Wikipedia.



(Fonte: Próprio autor.)

Ao utilizar as cinquenta amostras para realizar o treinamento no *software* GSA obteve-se como resultado que o mesmo era similar ao SDK de nome "bestallsharp", demonstrado na Figura 19.

Figura 19 – Exemplo de CAPTCHA referente ao SDK Bestallsharp.



(Fonte: Próprio autor.)

Este SDK possui uma acurácia de 56,86%, entretanto, ao tentar utilizar este SDK para a quebra, apenas 43,13% deles estavam corretos. Cada CAPTCHA levou cerca de 5 segundos para ser quebrado.

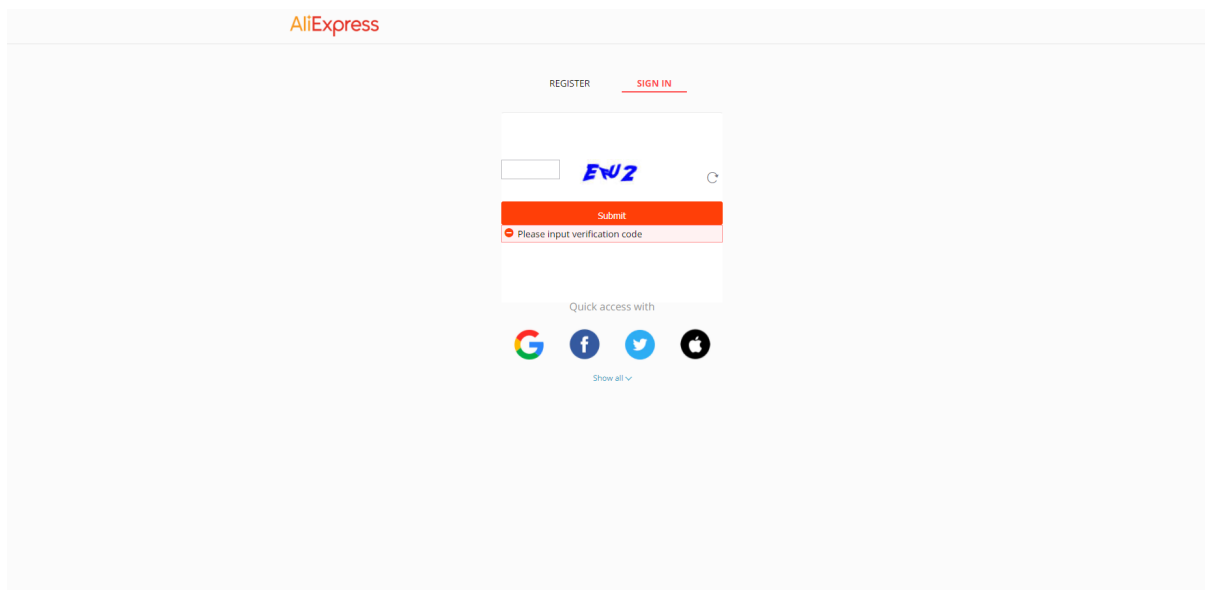
Ao realizar o procedimento descrito na Seção anterior sobre *services*, obteve-se 100% de respostas corretas com uma média de tempo de oito segundos e gastando um total de \$0,03 (três centavos de dólar).

Já a quebra via inteligência artificial, segundo o artigo [Ye et al. 2018], obteve em média, 67% de acurácia utilizando cerca de 4 milissegundos para cada CAPTCHA.

3.1.2 CAPTCHA AliExpress

O segundo CAPTCHA foi retirado do site *AliExpress*, um dos maiores comércios digitais do mundo, possuindo um tráfego mensal de 43,3 milhões de acessos [Farfan 2019], este CAPTCHA também é apresentado na tela de login do site (Figura 20), no intuito de proteger o site de tentativas de login automatizadas, área qual o consumidor acessa para realizar compras e verificar o andamento de pedidos anteriores. O CAPTCHA é de idealização e uso próprio.

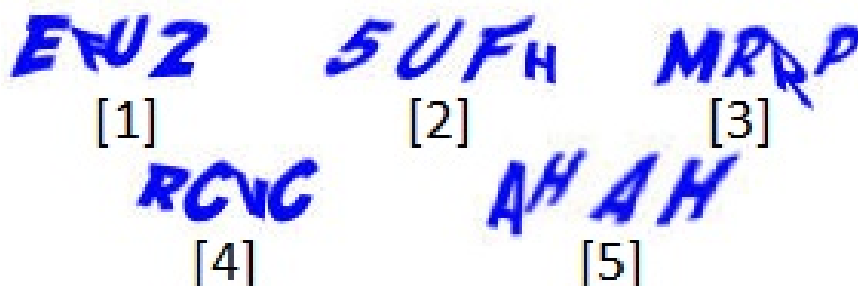
Figura 20 – Website AliExpress apresentando um desafio de CAPTCHA.



(Fonte: <https://login.aliexpress.com/>, acesso em 28/09/2020.)

Também utilizando cinquenta amostras de CAPTCHAs para a realização da análise, cada um possui um tamanho diferente, variando de 209 à 275 para a largura e de 48 a 72 para a altura. Possui exatamente quatro letras ou números. Tal qual o primeiro CAPTCHA, apresenta a aglomeração de caracteres. E possui leve distorção e uma leve angulação das letras, causando mais dificuldade para o reconhecimento dos contornos do caractere, conforme cinco exemplos destes CAPTCHAs demonstram na Figura 21.

Figura 21 – Exemplos de 5 CAPTCHAs do site da AliExpress.

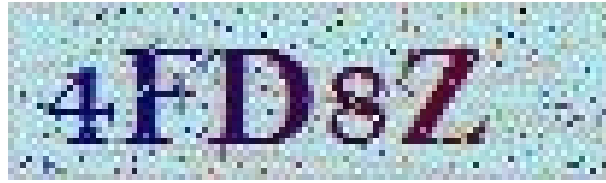


(Fonte: Próprio autor.)

Utilizando as cinquenta amostras para o treinamento no GSA obteve-se o SDK "HarveyDong" como similar (Figura 22).

Este SDK possui uma acurácia de 20%, entretanto, ao tentar utilizar este SDK para a quebra, apenas 6% deles estavam corretos. Cada CAPTCHA levou cerca de 3 segundos para ser quebrado.

Figura 22 – Exemplo de CAPTCHA referente ao SDK HarveyDong.



(Fonte: Próprio autor.)

Tal qual realizado para o CAPTCHA da Wikipédia, foram enviadas cinquenta requisições, sendo uma por cada CAPTCHA, ao final obteve-se 100% de respostas corretas com uma média de tempo de seis segundos e gastando um total de \$0,04 (quatro centavos de dólar).

E baseando-se no artigo [Ye et al. 2018], ao final do processo, cada CAPTCHA levava em média 3,7 milésimos de segundo e uma taxa de acurácia de 61%.

3.2 Quebra de reCAPTCHAs

A resolução dos desafios criados pelo Google serão efetuados de acordo com a possibilidade de quebra dos mesmos, ou seja, serão utilizados os métodos citados na Seção inicial deste capítulo (3) quando cabíveis.

3.2.1 v1

Conforme descrito na Seção 2.3.2.1, o reCAPTCHA V1 pode ser classificado como um esquema de CAPTCHA textual, então será tratado como tal;

Conforme citado, o serviço de reCAPTCHA V1 não está mais ativo, não sendo possível testar o desafio diretamente. Entretanto, obtendo imagens da internet e realizando um tratamento que consistia em recortar as bordas e informações do reCAPTCHA obtendo apenas o desafio, além de recortar o mesmo em dois, para facilitar o reconhecimento da imagem. Procedimento demonstrado na Figura 23

Ao fim do processo, foram obtidas duas imagens, as quais foram submetidas a um procedimento similar ao descrito na Seção 2.2, com o diferencial de que para o CAPTCHA ser considerado quebrado ambas as partes devem ser respondidas corretamente.

Inserindo cinquenta amostras obtidas desta forma, totalizando cem imagens, no GSA o mesmo reconheceu o SDK corretamente como sendo "reCAPTCHA". Entretanto não foi possível quebrar corretamente nenhuma das amostras inseridas, em 46% dos casos, apenas uma das imagens foi reconhecida corretamente, nos outros 54% nenhuma das imagens foi reconhecida. Obteve-se então uma acurácia de 0% enquanto a acurácia estimada do SDK era de 1%.

Figura 23 – Exemplo do tratamento realizado para a obtenção de imagens de um reCAPTCHA V1.



(Fonte: Próprio autor.)

Ao realizar o procedimento descrito na Seção 3 sobre *services*, obteve-se 98% de respostas corretas, demonstrando que até mesmo seres humanos estão passíveis de cometer erros quando um CAPTCHA é muito grande, conforme citado no artigo [Yan e Ahmad 2008]. A média de tempo para a resolução de cada CAPTCHA foi de onze segundos, gastando um total de \$0,06 (seis centavos de dólar).

Já a quebra utilizando inteligencia artificial citada no artigo [Ye et al. 2018] foi feita utilizando reCAPTCHAs obtidos em 2011 e alcançou uma média de 87,4% de acurácia, o tempo médio de quebra não foi citado.

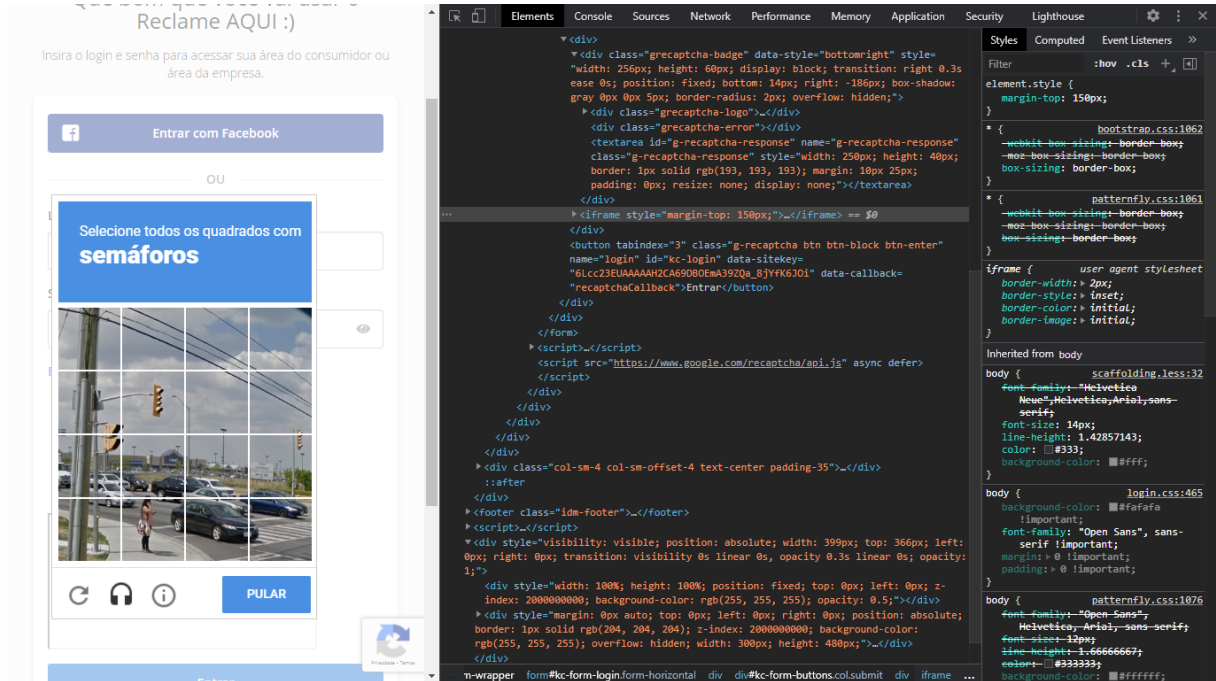
3.2.2 v2

Não foi encontrado nenhum *software* comercial de quebra para realizar a resolução do reCAPTCHA V2, então serão demonstradas apenas as quebras via *service* e via inteligencia artificial.

Para se resolver um reCAPTCHA V2 via *service* é preciso adotar uma abordagem diferente da utilizada para CAPTCHAs textuais, para tal foi necessário obter informações referentes ao desafio ao invés de uma imagem do mesmo. Essas informações são: A URL do site, endereço onde o desafio está sendo exibido naquele momento. E a "*WebsiteKey*", um código que se refere a aquele desafio em específico e pode ser obtido no código fonte da página, como pode ser visto na Figura 24. Como resposta se obtém um *token*, um código que deve ser inserido no código fonte da pagina no elemento comumente nomeado "recaptcha-token", porém o mesmo pode variar de acordo com a implementação do site, também demonstrado na Figura 24. Ao realizar cinquenta requisições ao *Anticaptcha*

contendo as informações citadas, obteve-se uma acurácia de 100% e um tempo médio de resolução de dezesseis segundos para cada desafio.

Figura 24 – Site do Reclame Aqui e o respectivo código fonte ao lado.



(Fonte: <https://www.reclameaqui.com.br>)

Conforme demonstrado no artigo [Sivakorn, Polakis e Keromytis 2016] não existem soluções ótimas para a quebra de reCAPTCHAs via inteligência artificial, sendo necessário treina-la para cada conjunto de imagens que o desafio apresenta, além de que tais imagens são constantemente atualizadas para melhorar a segurança do desafio. O mais próximo possível de um resultado obtido sem a interação humana é via *services* que funcionam a base de OCR. Como exemplo de tal método foram coletados dados do site do *service* "AZCaptcha" [AZCaptcha 2020].

Este *service* não detalha como o processamento é feito, apenas explicando o uso de OCRs para a quebra. Realizando uma simulação, foi possível estimar um gasto de \$0,09 (nove centavos de dólar), caso cinquenta reCAPTCHAs fossem submetidos, tal qual descrito na Seção anterior. Com um tempo estimado em oitenta segundos para cada reCAPTCHA.

3.2.3 v3

Como o desafio só é mostrado ao usuário caso a navegação seja detectada como irregular o foco da quebra deste CAPTCHA seria não ser reconhecido como um robô. Uma vez que o desafio é mostrado ele funciona tal qual um reCAPTCHA V2.

Devido a semelhança com a versão antecessora, não possuindo *software* comercial

de quebra, e devido a quebra via *service* sendo exatamente igual, tanto em valor quanto em velocidade e desempenho, a mesma não será demonstrada.

Devido ao recente lançamento desse desafio, não existem ainda estudos eficazes em sua quebra via inteligência artificial, existindo apenas trabalhos teóricos que iniciam a discussão e possibilitam implementações futuras devidamente eficientes, como pode ser visto no artigo [Akrou, Feriani e Akrou 2019]. Em tal estudo foi implementado uma forma de evitar que o desafio seja mostrado durante a navegação, já que a quebra é idêntica a do reCAPTCHA V2, o algoritmo que tenta simular a navegação via mouse de forma mais similar a de um humano, evitando que seja detectado como um acesso automatizado, entretanto, o ambiente simulado considera apenas um cenário ideal, existindo falhas em sua implementação no mundo real.

3.3 Quebra de FunCAPTCHAs

Tal qual descrito para reCAPTCHAs V2, não foi possível encontrar um *software* comercial para ser implementado o teste de quebra. Também não foi possível encontrar um documento científico descrevendo a quebra via inteligência artificial, apenas implementações amadoras sem documentação adequada, então este método não será demonstrado. Sendo descrita apenas a quebra via *service*.

Foram submetidos cinquenta FunCAPTCHAs ao Anticaptcha, utilizando o mesmo tipo de submissão descrito na seção de reCAPTCHA V2 (3.2.2). Ao fim do processo, obteve-se uma taxa de 98% de acurácia e o tempo médio de trinta segundos por CAPTCHA. Gastando-se o total de \$0,12 (doze centavos de dólar).

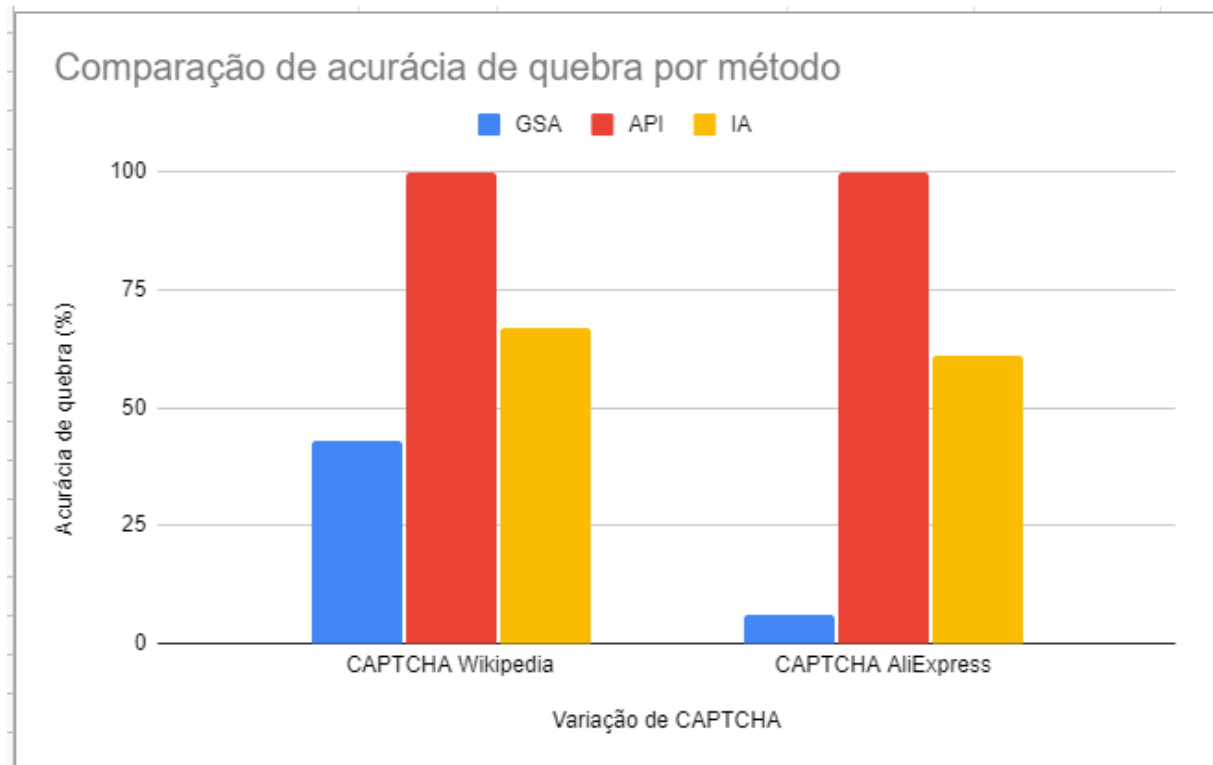
4 RESULTADOS

Neste capítulo a avaliação dos métodos de quebra serão medidos nos quesitos previamente citados no capítulo 3, sendo eles tempo e acurácia. Serão descritos os resultados obtidos das quebras dos CAPTCHAs textuais, reCAPTCHAs e FunCaptchas. Além de demonstrar uma comparação entre os métodos de quebra entre si.

4.1 Resultados CAPTCHAs Textuais

Fazendo uma comparação entre as metodologias de quebra para CAPTCHAs textuais (vide Seção 3.1) é possível concluir que o resultado ótimo em relação a acurácia é dada sempre por seres humanos, ou seja, *services* possuem o melhor desempenho nesse quesito, conforme pode ser visto no gráfico da Figura 25.

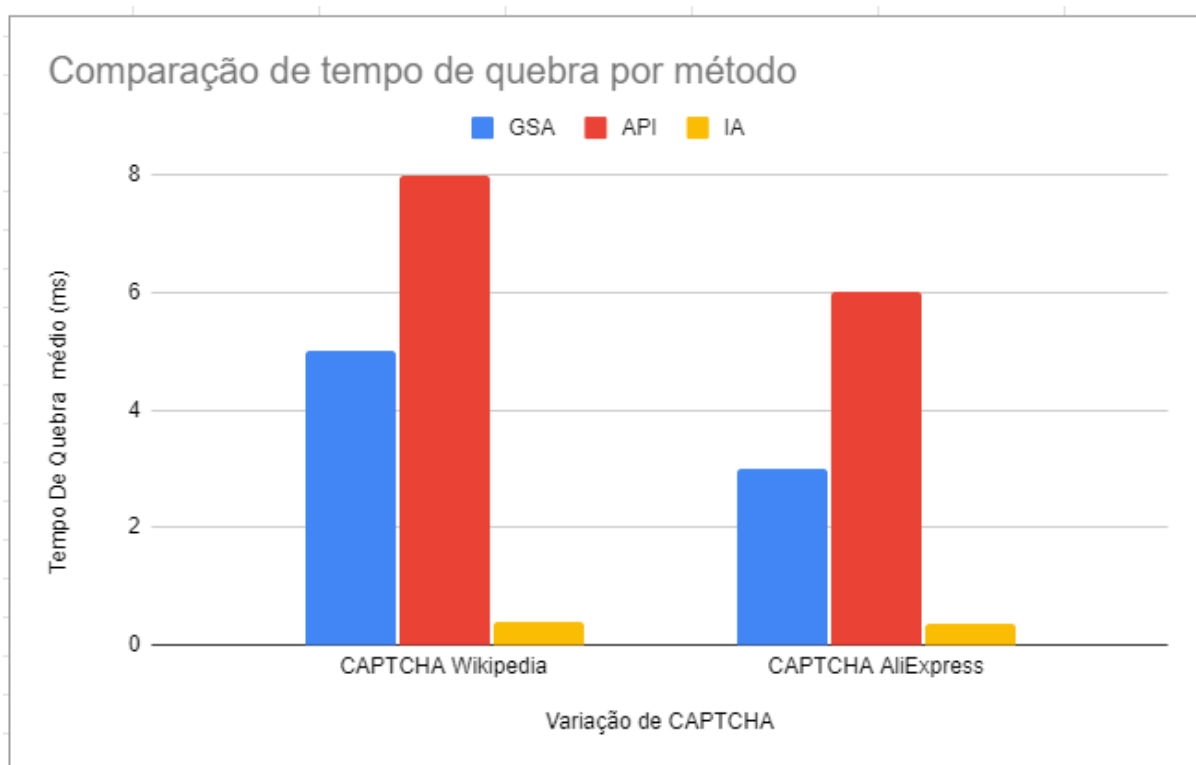
Figura 25 – Gráfico de comparação de acurácia entre métodos de quebra para CAPTCHAs textuais.



(Fonte: Próprio autor.)

Entretanto, no quesito velocidade de quebra, é discrepante a velocidade em que a inteligência artificial consegue processar o CAPTCHA quando comparado com outros métodos de quebra, vide Figura 26.

Figura 26 – Gráfico de comparação de tempo entre métodos de quebra para CAPTCHAs textuais.



(Fonte: Próprio autor.)

De um modo geral, pode se concluir que o desempenho do *software* de quebra foi satisfatório para quebras textuais, principalmente tendo em vista que o mesmo necessita apenas da compra de uma licença vitalícia, enquanto *services* e inteligências artificiais necessitam de um investimento maior e contínuo. Sendo necessário pagar uma quantia em dinheiro ao *service* sempre que uma quebra for realizada e aplicar um novo esforço ao treinamento da rede neural sempre que um novo esquema de CAPTCHA for criado ou melhorado.

Porém não é necessário escolher apenas um dos métodos, é possível realizar uma combinação deles. Como por exemplo: Enviar o CAPTCHA para ser quebrado no software, que seria a alternativa mais barata a longo prazo, e então caso o mesmo falhasse repetidas vezes, enviar o mesmo a um *service*, alternativa com maior acurácia. Desta forma, otimiza-se o gasto financeiro com a quebra, em detrimento do tempo gasto.

Em suma, é necessário escolher qual a característica mais importante para a quebra em detrimento das outras. Uma quebra mais rápida pode também ser mais custosa e menos acurada, enquanto uma quebra que leva mais tempo pode economizar recursos financeiros e obter uma melhor acurácia.

4.2 Resultados reCAPTCHA

Já na comparação entre a segurança das versões do reCAPTCHA é notável o avanço de tecnologia aplicado, tendo em vista que atualmente não existem softwares comerciais disponíveis para as novas versões e nem inteligências artificiais capazes de solucionar os mesmos. Sendo necessário recorrer a *services*, sejam eles com a força de trabalho baseada em humanos ou baseados em OCR. Neste caso, a solução ótima para realizar a navegação automática em tais sites seria evitar que o desafio fosse mostrado e caso o mesmo fosse proposto recorreria-se ao uso de *services*.

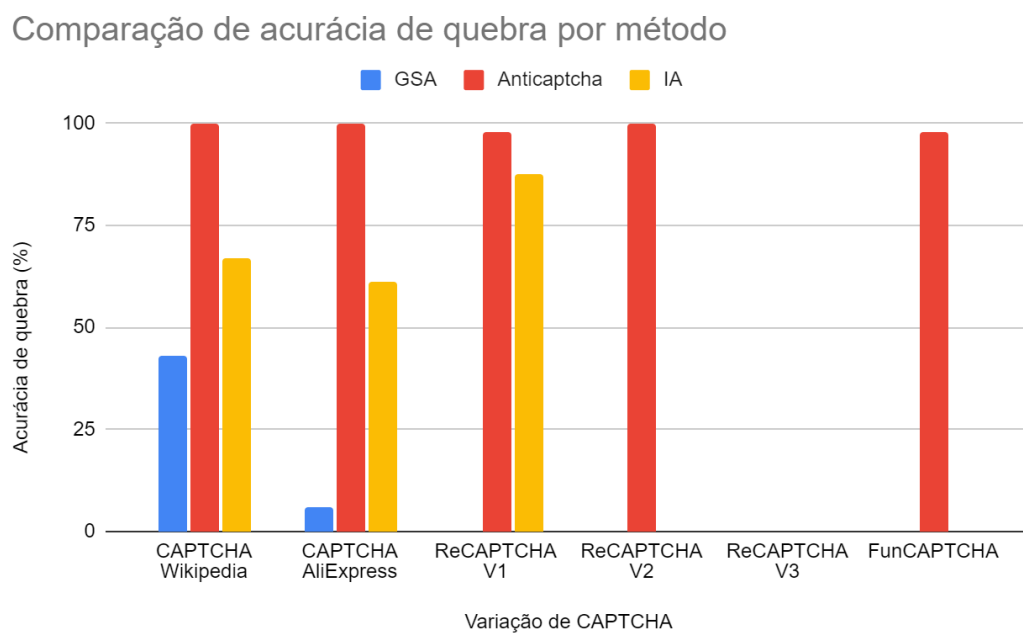
4.3 Resultados FunCAPTCHA

O desafio intitulado FunCAPTCHA obteve um bom nível de segurança quando comparado com outros tipos. Não possuindo software capaz de quebra-lo ou inteligência artificial que realize essa tarefa satisfatoriamente. Sendo necessário recorrer a *services*, tanto os baseados em humanos ou baseados em OCR.

4.4 Resultados Gerais

Em geral, é possível concluir que dentre os métodos de quebra o que mais se destaca em flexibilidade são os *services*, possuindo opções para CAPTCHAs textuais, reCAPTCHAs e FunCaptchas.

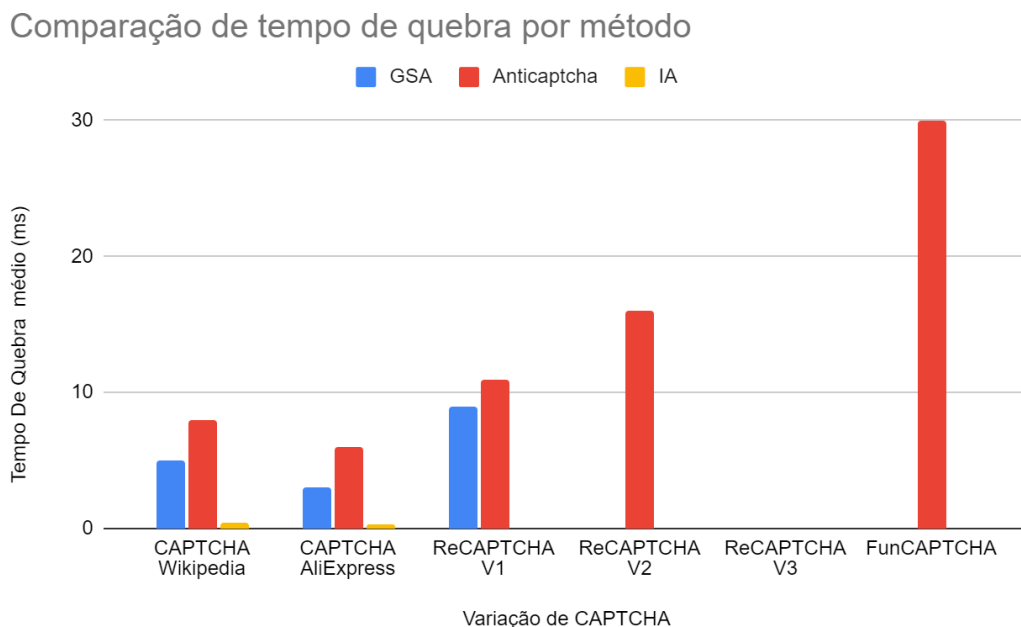
Figura 27 – Gráfico de comparação de acurácia entre métodos de quebra.



(Fonte: Próprio autor.)

E em todos os CAPTCHAs testes possuí uma alta acurácia (Figura 27) e tempo razoável (Figura 28), tendo como desvantagem apenas o custo a longo prazo.

Figura 28 – Gráfico de comparação de tempo entre métodos de quebra.



(Fonte: Próprio autor.)

Além disso, exceto o *service* nenhum dos outros métodos de quebra possuem suporte eficaz e sustentável a qualquer outro tipo de CAPTCHA não textual, ficando defasados com a constante migração e melhoramento das técnicas de segurança, algo notável até mesmo no artigo de inteligência artificial em que este trabalho foi embasado, diversos dos sites citados migraram para reCAPTCHA ou até mesmo criaram seu próprio desafio, como por exemplo, o *Ebay*. Tendo em vista esta constante melhora, outros métodos de quebra necessitam de demasiada manutenção, enquanto os *services* não. Sendo necessário manutenção do acesso automatizado apenas quando a estrutura do site em que o acesso está sendo realizado é alterada.

Enquanto isso, os softwares são uma alternativa relativamente razoável apenas quando se trata de CAPTCHAs textuais, sendo ineficazes em reCAPTCHAs e FunCaptchas. Por possuir um custo apenas imediato, com a compra da licença vitalícia, e nenhum custo de manutenção são uma boa alternativa a longo prazo. Entretanto não são um método garantidamente eficaz, tendo seu desempenho diretamente dependente do desafio textual que se pretende quebrar.

Já a inteligência artificial tal qual os *softwares* acabam demonstrando melhores resultados em CAPTCHAs textuais, sendo menos eficazes em outros tipos. Entretanto, sua viabilidade é muito baixa para o uso pelo público comum, sendo necessário possuir um conhecimento maior na área da ciência da computação para que haja uma implementação

de quebra. Entretanto, para fins acadêmicos, são um ótimo meio para testar a segurança, vulnerabilidades e pontos fortes de um CAPTCHA.

5 CONCLUSÃO

Este trabalho teve como objetivo realizar um levantamento geral sobre os principais CAPTCHAs presentes no cotidiano dos usuários da internet, além de demonstrar os métodos de quebra para CAPTCHAs textuais, reCAPTCHAs e FunCaptchas e fazer uma comparação entre tais métodos. Com o auxílio de trabalhos co-relatos, dados empíricos e informações obtidas dos websites dos *services* foi possível avaliar e demonstrar que apesar de não ser uma tarefa trivial é possível quebrar CAPTCHAs com uma certa facilidade e concluir que os mesmos, apesar de seguros, possuem brechas de segurança.

Foi demonstrado que existem diversas formas de se quebrar um CAPTCHA, entretanto, existe sempre uma mais adequada para cada caso, não existindo uma alternativa ótima em todos os quesitos que foram avaliados, como o tempo, a acurácia e o custo, para todas as variações de CAPTCHA. Por exemplo, pode-se usar um software de quebra para CAPTCHAs textuais, porém, não para reCAPTCHAs, FunCAPTCHAs e demais variações. Como foi demonstrado na Seção 3, os *services* são uma alternativa altamente eficaz, principalmente os que utilizam-se de esforço humano para a quebra.

Apesar de serem um dispositivo de segurança válido e fornecerem um certo nível de proteção, essa defesa pode ser burlada, caso o indivíduo empregue um nível de esforço maior. Necessitando buscar a cada dia o aprimoramento dos CAPTCHAs e seus dispositivos de segurança.

Como demonstrado o reCAPTCHA, atualmente, possui os mecanismos de segurança mais avançados, porém ainda assim é possível burlar os mesmos com o auxílio de *services*, indicando que não existe um CAPTCHA que não possa ser burlado ou que não possua algum tipo de falha de segurança.

Também é importante ressaltar que este trabalho não possui o intuito de incentivar qualquer tipo de prática ilícita que possa decorrer da quebra de CAPTCHAs, sendo apenas um estudo objetivo e imparcial de como tais dispositivos funcionam e podem ser burlados.

Trabalhos futuros

Para trabalhos futuros tem-se como objetivo as seguintes proposições:

1. Análise da viabilidade da quebra de reCAPTCHAs utilizando redes neurais;
2. Demonstração da mutabilidade e avanços dos CAPTCHAs ao longo dos anos;

Referências

- 2CAPTCHA. *Bypass Captcha*. 2020. Disponível em: <<https://2captcha.com>>.
- AHN, L. von; BLUM, M.; LANGFORD, J. *Telling Humans and Computers Apart Automatically*. COMMUNICATIONS OF THE ACM. Volume 47, Número 2, 2004. 2–3 p. Disponível em: <<https://iopscience.iop.org/article/10.1088/1757-899X/245/1/012001/pdf>>.
- AKROUT, I.; FERIANI, A.; AKROUT, M. *Hacking Google reCAPTCHA v3 using Reinforcement Learning*. 2019. Disponível em: <<https://arxiv.org/pdf/1903.01003.pdf>>.
- ANTICAPTCHA. *Anti API v.2 Documentation*. 2020. Disponível em: <<https://anticaptcha.atlassian.net/wiki/spaces/API/pages/196635/Documentation+in+English>>.
- ANTICAPTCHA. *Anti Captcha Overview*. 2020. Disponível em: <<https://anti-captcha.com/mainpage>>.
- ARICA, N.; VURAL, F. T. Y. *Optical character recognition for cursive handwriting*. 2012. 2–3 p. Disponível em: <<https://iopscience.iop.org/article/10.1088/1757-899X/245/1/012001/pdf>>.
- AZCAPTCHA. *AZCaptcha*. 2020. Disponível em: <<https://azcaptcha.com>>.
- Captcha Sniper. *Captcha Sniper(CS)*. 2020. Disponível em: <<https://www.captchasniper.com/>>.
- CHOW, Y. W.; SUSILO, W. *Text-based CAPTCHAs over the years*. IOP Conf. Series: Materials Science and Engineering 273 (2017) 012001, 2017. 3–4 p. Disponível em: <<https://iopscience.iop.org/article/10.1088/1757-899X/245/1/012001/pdf>>.
- ENDCAPTCHA. *EndCaptcha*. 2020. Disponível em: <<https://www.endcaptcha.com>>.
- FARFAN, B. *The Biggest Internet and Mobile Retail Shopping Sites*. 2019. Disponível em: <<https://www.thebalancesmb.com/top-mobile-retail-shopping-sites-2891929>>.
- FERRANTE, T.; FERRANTE, C. M. *E-Leetspeak: All New! the Most Challenging Puzzles Since Sudoku!*. 2008. 3 p.
- GAO, H. et al. *The Robustness of Hollow CAPTCHAs*. 2013. Disponível em: <<https://prof-jeffyan.github.io/ccs13.pdf>>.
- KERAS. *Keras Python*. 2020. Disponível em: <<https://keras.io/about>>.
- LABS, F.-S. *CAPTCHA-22: Breaking Text-Based CAPTCHAs with Machine Learning*. 2019. Disponível em: <<https://labs.f-secure.com/blog/captcha22>>.
- MOREIN, W. G. et al. *Using Graphic Turing Tests To Counter Automated DDoS Attacks Against Web Servers*. Tenth ACM Conference on Computer and Communications Security, 2003. 8–19 p. Disponível em: <<https://dl.acm.org/doi/10.1145/948109.948114>>.

- NOURY, Z.; REZAEI, M. *Deep-CAPTCHA: a deep learning based CAPTCHA solver for vulnerability assessment*. 2020. Disponível em: <<https://arxiv.org/pdf/2006.08296.pdf>>.
- OPENCV. *OpenCV About*. 2020. Disponível em: <<https://opencv.org/about>>.
- PYTORCH. *PyTorch Tutorials*. 2020. Disponível em: <<https://pytorch.org/tutorials>>.
- SALVATORE, F. *Deep learning drops : breaking captcha*. 2017. Disponível em: <<https://towardsdatascience.com/deep-learning-drops-breaking-captcha-20c8fc96e6a3>>.
- Sebastian McKenzie. *Insomnia*. 2020. Disponível em: <<https://insomnia.rest>>.
- SIVAKORN, S.; POLAKIS, J.; KEROMYTIS, A. D. *I'm not a human: Breaking the Google reCAPTCHA*. 2016. Disponível em: <<https://www.blackhat.com/docs/asia-16/materials/asia-16-Sivakorn-Im-Not-a-Human-Breaking-the-Google-reCAPTCHA-wp.pdf>>.
- SOLVER, B. C. *Best Captcha Solver*. 2020. Disponível em: <<https://bestcaptchasolver.com>>.
- Sven Bansemer. *GSA Captcha Breaker*. 2020. Disponível em: <https://www.gsa-online.de/product/captcha_breaker>.
- TENSORFLOW. *Introdução ao TensorFlow*. 2020. Disponível em: <<https://www.tensorflow.org/learn>>.
- TYPERZ, I. *Image Typerz*. 2020. Disponível em: <<https://www.imagetyperz.com>>.
- WIKIPEDIA. *Wikipedia:About*. 2020. Disponível em: <<https://en.wikipedia.org/wiki/Wikipedia:About>>.
- YAN, J.; AHMAD, A. S. E. *Usability of CAPTCHAs Or usability issues in CAPTCHA design*. 2008. Disponível em: <<https://core.ac.uk/download/pdf/194585355.pdf>>.
- YE, G. et al. *Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach*. 2018. Disponível em: <<https://eprints.lancs.ac.uk/id/eprint/126984/1/ccs18.pdf>>.
- ZHAO, N.; LIU, Y.; JIANG, Y. *CAPTCHA Breaking with Deep Learning*. 2017. Disponível em: <<http://cs229.stanford.edu/proj2017/final-reports/5239112.pdf>>.