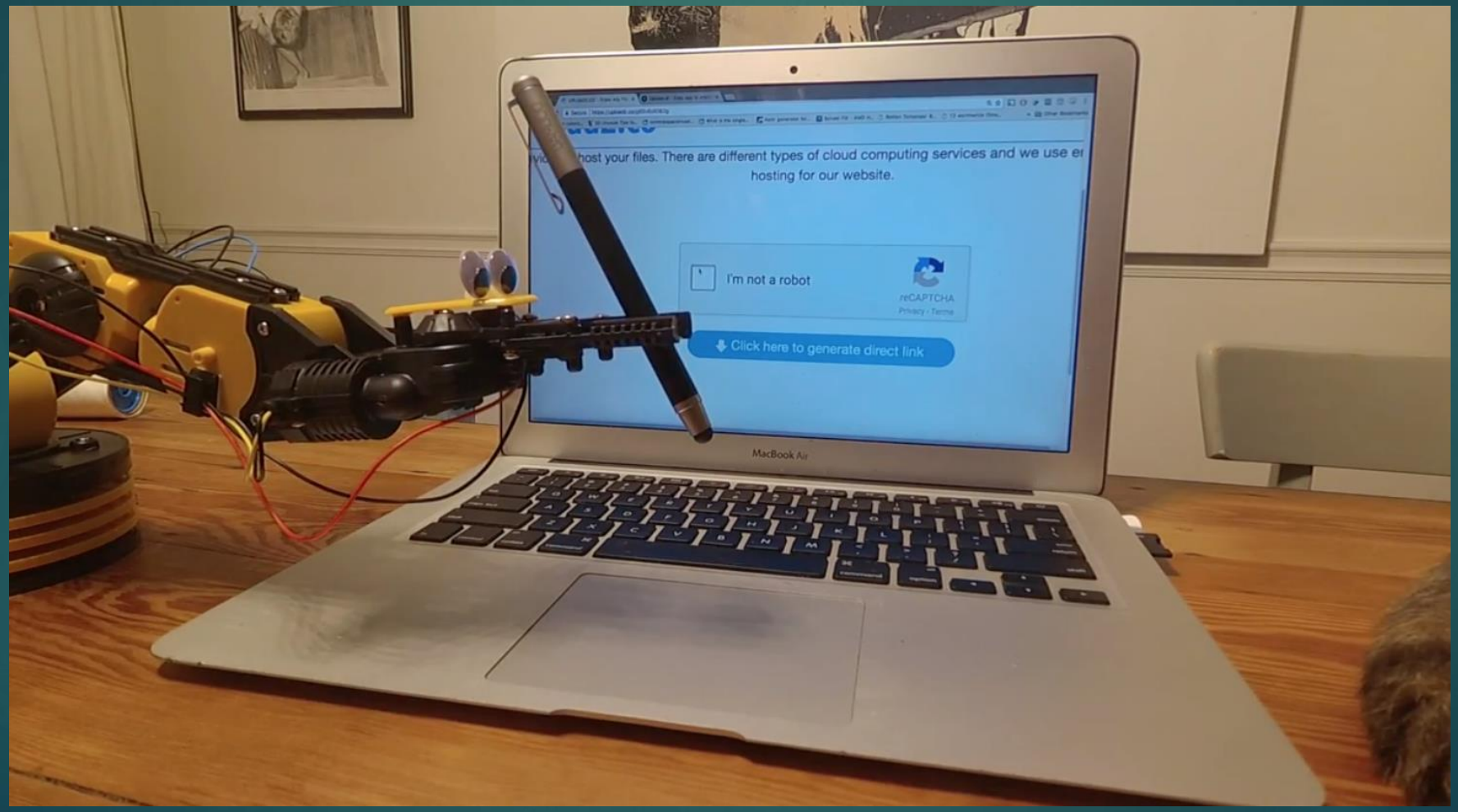
# unCaptcha

A LOW-RESOURCE DEFEAT OF RECAPTCHA'S ÁUDIO CHALLENGE

ALUNO: PATRIK BEDIN SCHETTERT

# O que é CAPTCHA?



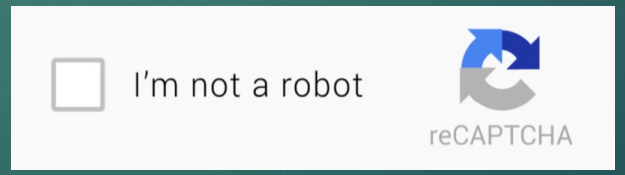
# O que é CAPTCHA?

- Completely Automated Public Turing tests to tell Computers and Humans Apart
- Sistema de segurança utilizado para proteger contra criação de contas automatizada e abuso de serviço
- Funciona a partir de desafios lançados ao usuário( fácil para humanos mas difícil para computadores)
- Bots e Sybil atacks
- Plataformas exigem a resolução de um captcha para criação de registro.
- Burlar sistemas de captchas pode teoricamente permitir o controle de fluxo de informações em massa
- Devido a sua importância, esse sistema tem sido alvo de ataques por anos

# Tipos de CAPTCHA

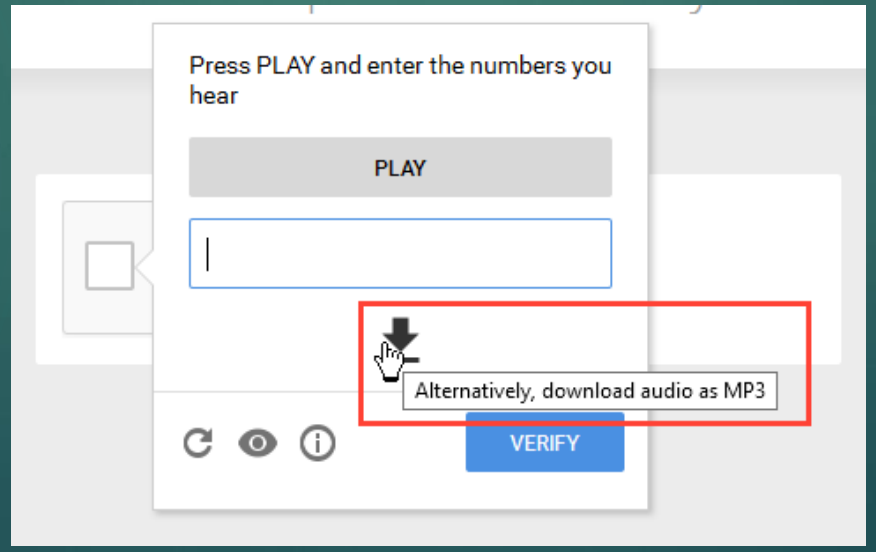
- Texto retorcido (2009- 2014)
  - Utilizados até hoje
  - Provados serem inseguros (98% de eficácia na resolução)


- Imagem (2014 – atualmente)
  - Em substituição aos de texto retorcido




# Tipos de CAPTCHA

- Audio
  - Especialmente desenvolvido para usuários com algum tipo de deficiência visual
  - Diferentes narradores, dizendo letras ou dígitos em intervalos randômicos, a velocidades variáveis, geralmente com sotaque ou distorção adicionada.



# Tentativas de burlar áudio CAPTCHA

- Geralmente, ataques a áudio captchas são realizados por meio do modelos de aprendizado de máquina treinados para identificar as palavras faladas (alta utilização de recursos e tempo)

- Serviços de reconhecimento de voz, como Sphinx ou Google Speech Recognition (não são precisos o suficiente para resolver os captchas).

- unCaptcha – sistema automatizado de ataque ao Google reCaptcha 2017, utilizando serviços públicos de reconhecimento de voz e realizando mapeamento fonético para aumentar a precisão.

# Ataques já realizados

- Kochanski et al. Investigou a segurança dos captchas e concluiu que os humanos tem grande superioridade no reconhecimento de áudio captchas, se comparados a sistemas de reconhecimento de voz automáticos, quando ruído é adicionado ao dígitos

- Primeiras tentativas (burlas Google e Yahoo) – tática de resolver em duas fases
  - Primeiramente são extraídos porções que contém os dígitos
  - Em sequencia cada digito é classificado individualmente por algoritmos de aprendizado de máquina previamente treinados

# Ataques já realizados

- Decaptcha eBay (2012) – analise dos picos de energia do áudio para separação de dígitos
  - Utilizava algoritmo de aprendizado supervisionado e um grande banco de dados de captchas do eBay
  - Acurácia de 75%
  - São necessários muitos recursos computacionais( núcleos, largura de banda, endereçs IP, etc...). 10% de sucesso em baixo/médio disponibilidade de recursos
- Defcon group  reCaptcha (2012)
  - Rede neural local treinada
  - Acurácia de 99%
  - Google atualizou o sistema trocando palavras para dígitos, adicionando ruído e variando o numero de dígitos ( 61% de acurácia)
  - Limitação era o fato de precisar de gigabytes de dados para o treinamento
  - 2016 uma nova abordagem utilizava o próprio reconhecimento de voz do google para resolver os captchas.
  - 2017 atualização recaptcha adicionou mais dígitos (10) e ruído de fundo em cada digito, inviabilizando o método anterior.

# Modelo de ameaça

- A maioria das tentativas de ataque consideram que o atacante possui bastante recurso
- Várias tentativas são efetuadas para um numero pequeno de acertos
- Isso em grande escala pode ser utilizado para abusar de serviços
- A acurácia suficiente para um ataque depende do numero de recursos
  - 1/10000 é suficiente para atacantes com muitos recursos
  - Para poucos recursos é considerado uma acurácia superior a 50%
  - unCaptcha considera apenas um computador, um IP, pouca memoria RAM e dados de treinamento limitados.

# Funcionamento do reCaptcha

- Trabalha utilizando uma engine de analise de riscos
- Utiliza o conceito de "nível de suspeita" determinado por meio de interações do usuário (cliques de botão e digitação).
- noCaptcha reCaptcha – quando o sistema é confiante que o usuário é humano
- Os desafios são nivelados de acordo com a suspeita do usuário.
- A suspeita é avaliada a partir do histórico de interação com serviços do Google.
- Harder challenge – 10 dígitos ou várias desafios com imagem.
- Google mantem em segredo como o sistema funciona.
- Roda em de forma criptografada, em uma VM isolada, utilizando JS e uma linguagem de bytecode única.

# Funcionamento do reCaptcha

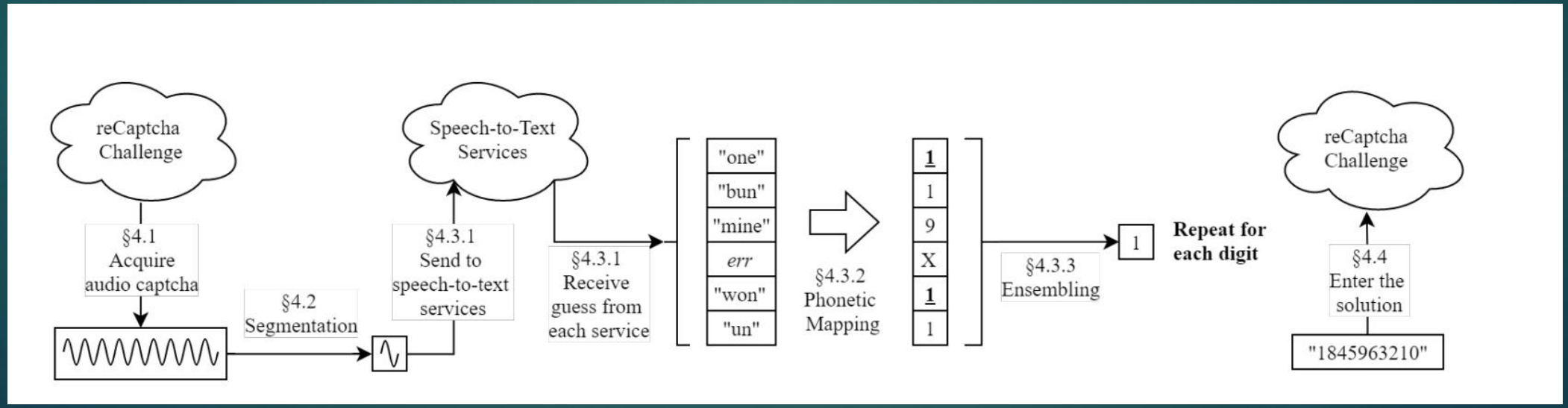
- O bytede tem acesso as variáveis JS do seu próprio interpretador
- A chave de descriptografia é alterada em diversos pontos durante a própria execução
- Um disasembler e descompilador foi desenvolvido e determinou que o captcha verifica a presença de:
  - Plugins válidos
  - String de user-agente válida
  - Resolução de tela válida
  - Tempo de execução
  - Teclado
  - Fuso horário
  - Cookies do servidor
  - Ações de toque no frame do captcha
  - Numero de cliques
  - Funções do navegador e regras de CSS
  - Propriedades de renderização do canvas

# Funcionamento do reCaptcha

- Em 2016 Sivakorn et al. Explorou fraquezas do sistema de captcha por imagem da Google.
- Foi descoberto que os tracking cookies da Google executavam um papel fundamental na defesa do captcha
- O sistema é ciente de todas as vezes que o usuário interage com um serviço do Google.
- Utilizando dessa informação, um sistema automatizado navegando por 9 dias por diferentes serviços do Google conseguiu enganar o sistemas de risco do captcha.
- Cada cookie pode completar automaticamente apenas 8 captchas por dia
- Tentativa de identificar bot antes do captcha. Problema com o desafio de áudio.

# unCaptcha - Etapas

- Obter a amostra de áudio
- Segmentar o áudio  partes menores (dígitos)
- Analisar cada segmento de áudio
- Inserir a solução do captcha

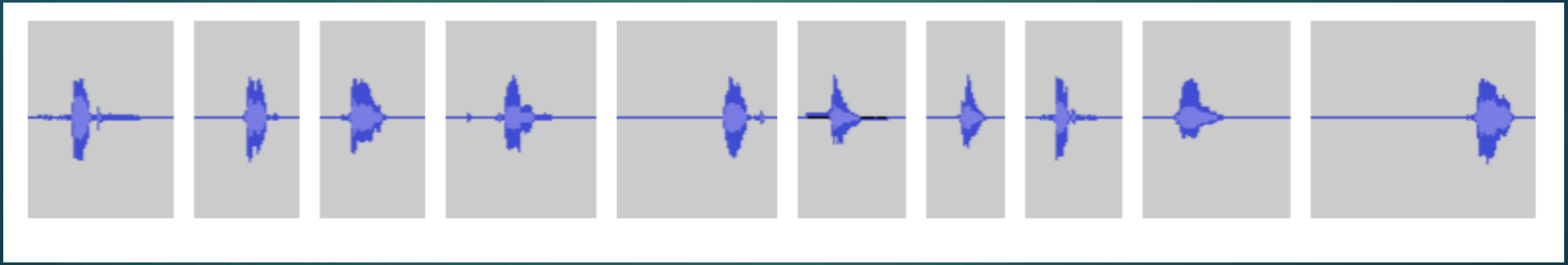


# Adquirindo audio

- reCaptcha utiliza múltiplas heurísticas para distinguir humanos de bots (navegador, headless browser, checagem de javascript, etc)
- Entre elas esta a verificação de movimentos de mouse e teclado "inorgânicos" (alta velocidade, linha reta)
- Passos realizados no site Reddit:
  - Selenium ( software de automação do navegador
  - unCaptcha procura um proxy HTTP para mascarar a conexão (Gather Proxy)
  - Navega até a pagina de cadastro, digita os dados com valores randômicos e espaço de tempo variável entre cada tecla.
  - Localiza o botão "Im not a robot" e clica nele
  - Bot modifica para "Audio challenge", faz o download do arquivo de áudio, e clica no botão de play para simular um usuário.

# Pre-processamento

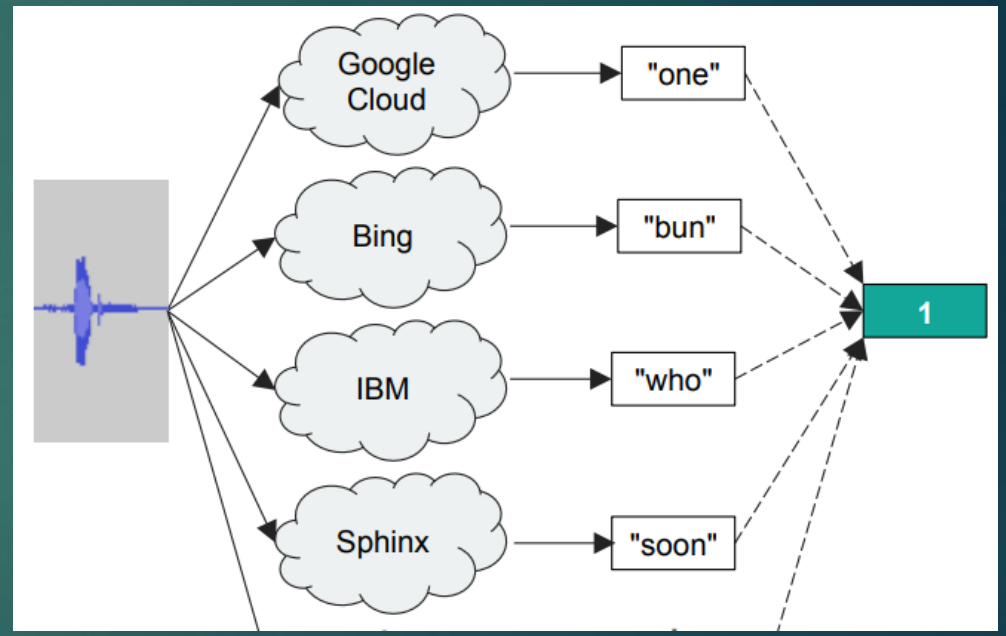
- Segmentação do áudio em partes menores
- reCaptcha possui distorções no áudio de cada segmento
- Isso facilida a separação dos dígitos
- unCaptcha simplesmente divide o áudio em períodos de silencio utilizando analise de amplitude.



# Analisando o audio

- Faz uso de ferramentas de reconhecimento de voz online existentes, com adição de mapeamento fonético e *essembling*.

- Serviços de conhecimento de voz
  - unCaptcha faz upload de cada digito para um conjunto de serviços de reconhecimento de voz( Google Cloud, Bing Speech Recognition, IBM Bluemix, Google Speech API, Wit-AI e Sphinx.
  - Cada um dos serviços retorna um candidado de texto transcrito.
  - Problema com confusão entre palavras e dígitos.

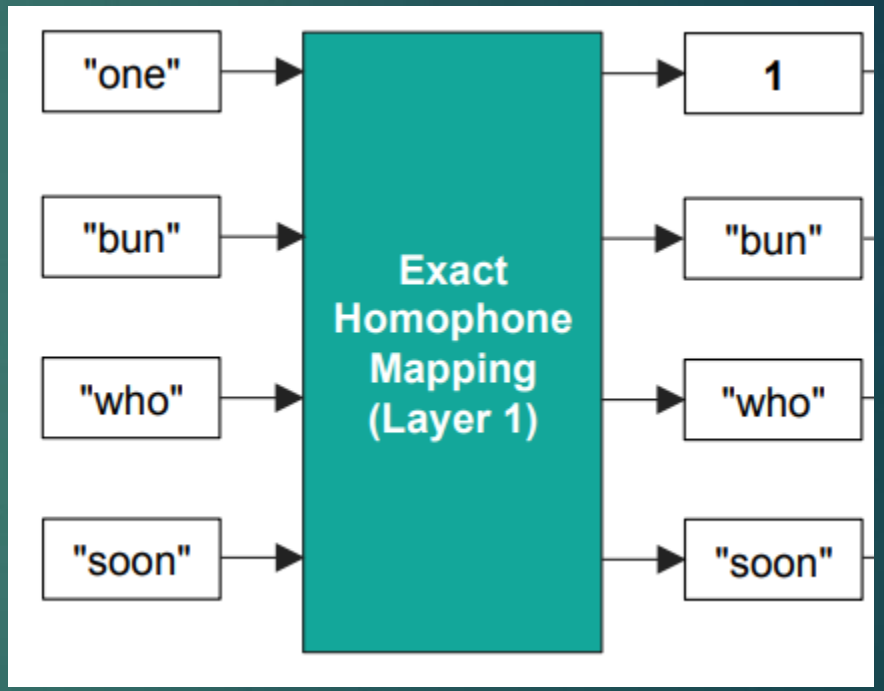


# Mapeamento fonético

- Duas camadas : homófonos e quase-homófonos
  - Homófonos
    - Camada utilizada para mapear erros comuns dos sistemas de reconhecimento de voz para seus respecitivos dígitos numéricos.

      Exemplo: to/too -> 2  for/fore -> 4.
    - Além disso, geralmente os dígitos são escritos por extenso, tendo sido mapeados os 10 dígitos para sua forma numérica

      Exemplo: four ao invés de 4

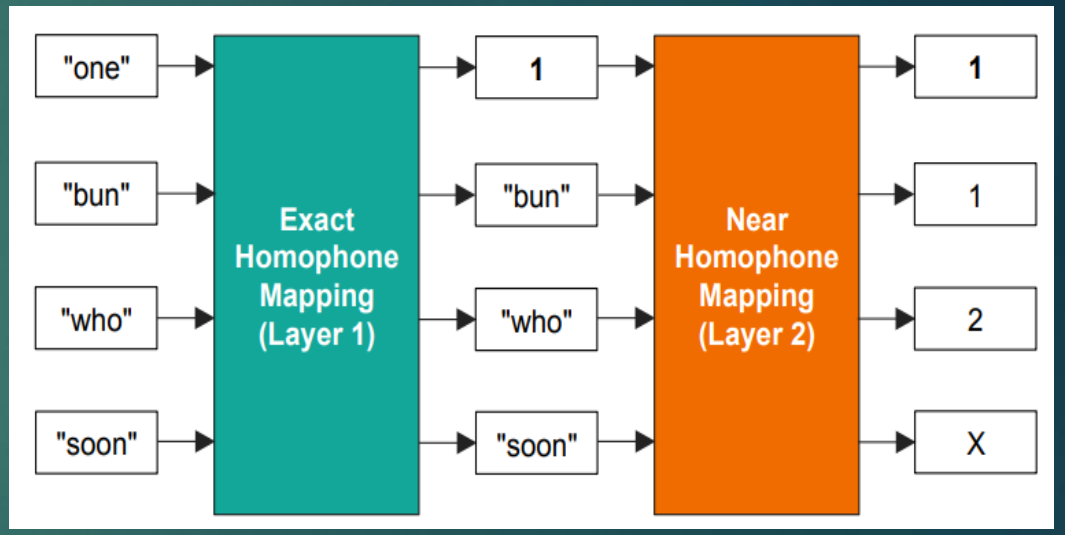


# Mapeamento fonético

- Duas camadas : homófonos e quase-homófonos
  - Quase-homófonos
    - Camada utilizada pare corrigir erros afetados pela distorção, sotaque acentuado e pequenos segmentos de áudio
    - Mapeamento baseado em heurística
      - Foram adicionados manualmente 36 quase-homófonos, resultando em uma acurácia maior que 50%

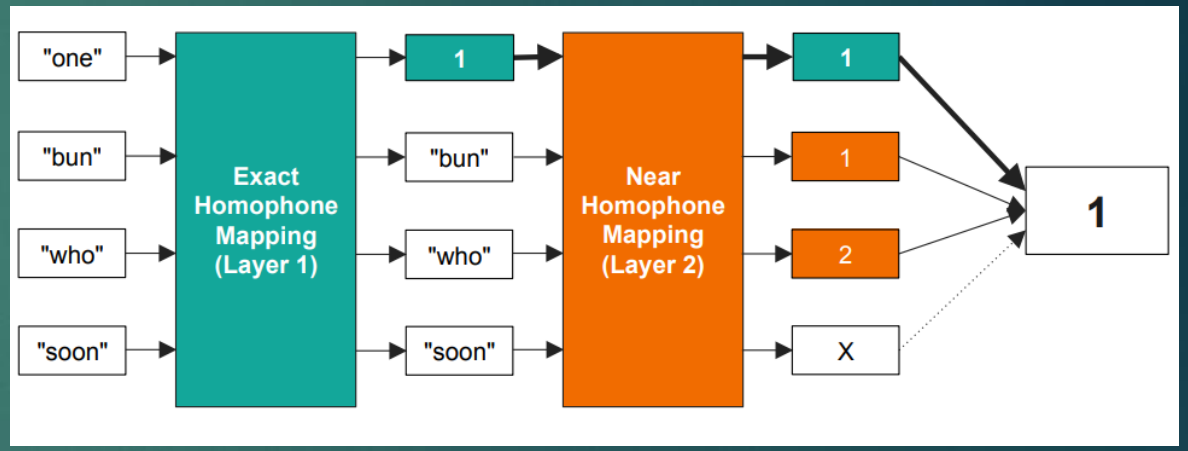
      Exemplo: free -> 3 sex -> 6 mine -> 9.
    - Adicionadas correspondências parcias

      Palavras terminadas em "icks" ->6; "ee" -> 3;
    - Ao todo compreende 129 homófonos, quase-homófonos e correspondecias parciais
    - Utilizado analise estatística para melhorar a identificação de outros mapeamentos
      - "you know" -> 0 (melhorou a taxa de sucesso em 0.6%)



# Essembling

- Cada candidato a resposta recebe um voto com determinado peso. A resposta com maior peso vence.
- Esse peso é atribuído de duas formas:
  - De acordo com o serviço de reconhecimento de fala( alguns são mais precisos do que outros)
  - Baseado na similaridade fonética de palavras específicas
    - Exemplos: “fine” -> 5 ou 9; “brain” -> 9; “we” -> “3”; “they’ve” -> 8
    - Too ou for -> 2 ou 4




- Essas alterações aumentaram a acurácia para mais de 65%

# Precisão por dígito

| Service | Per-digit Accuracy Base | + Mapping | Captcha Success |
|---|---|---|---|
| Google Cloud | 55.55% | 55.81% | 2.61% |
| Wit | 48.01% | 64.45% | 8.06% |
| Bing | 44.12% | 66.65% | 10.39% |
| IBM | 31.62% | 62.67% | 6.50% |
| Google | 21.25% | 28.10% | 0.01% |
| Sphinx | 6.64% | 31.88% | 0.02% |

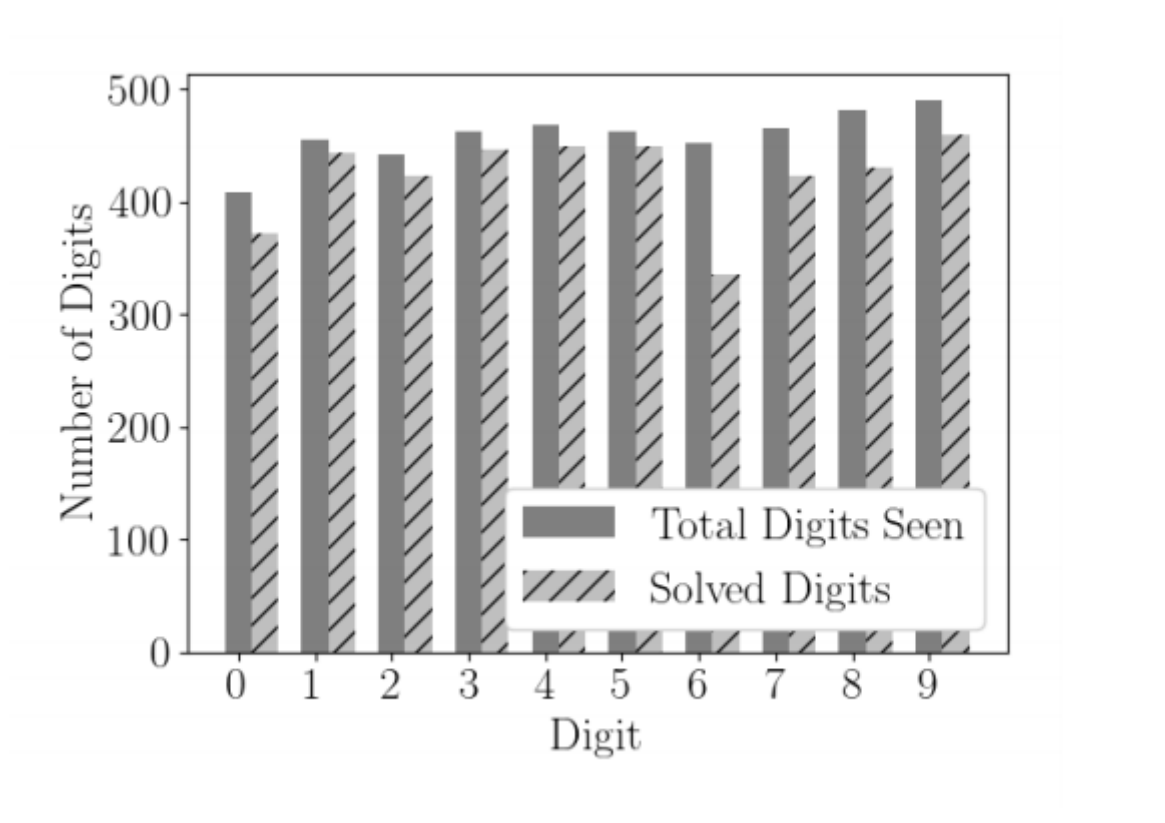
Precisão antes e depois de realizar o mapeamento fonético (desafio de 10 dígitos).

# Benefícios do "essembling"

| *N* | Best Ensemble of *N* Services | Per-digit Accuracy | Captcha Success |
|---|---|---|---|
| 1 | Google Cloud | 55.81% | 2.19% |
| 2 | Bing + IBM | 82.87% | 47.26% |
| 3 | + Google Cloud | 88.25% | 66.96% |
| 4 | + Wit | 91.36% | 78.12% |
| 5 | + Sphinx | 91.93% | 80.09% |
| 6 | + Google | 91.99% | 80.31% |
| | + "X" $\mapsto$ "6" | **93.41%** | **85.15%** |

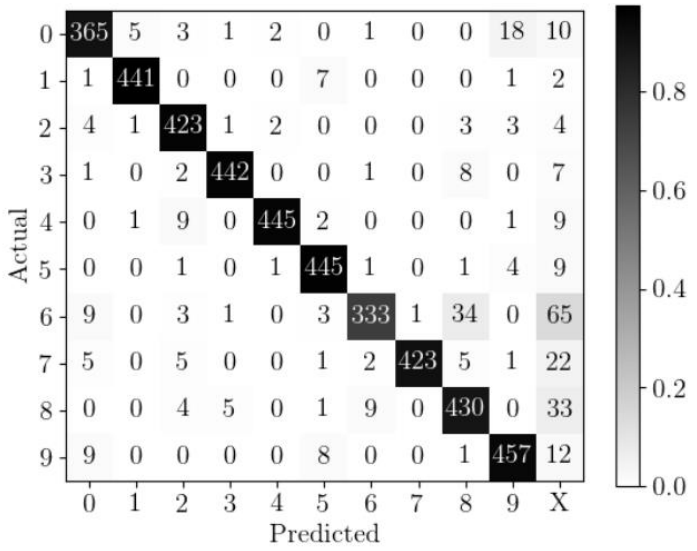
Benefício de considerar vários serviços de reconhecimento de voz, após efetuado o mapeamento fonético.

# Desempenho do unCaptcha

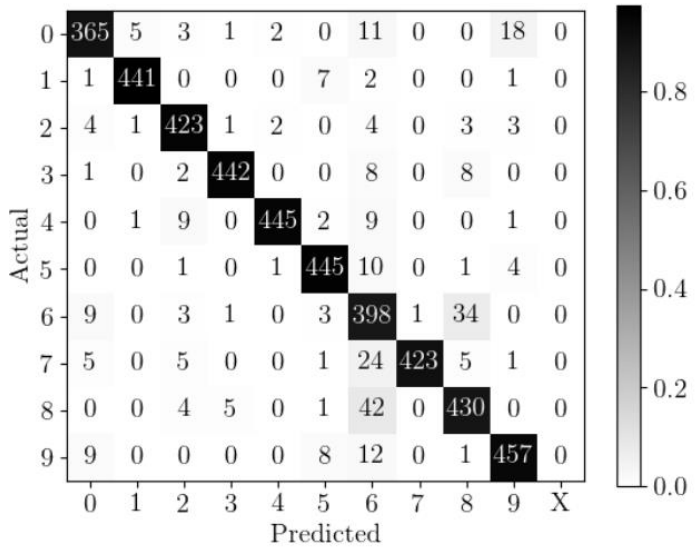



Distribuição dos dígitos recebidos pelo reCaptcha( praticamente uniforme).

# Precisão do número “6”




(a) *Without* mapping unknowns to “6”.




(b) *With* mapping unknowns to “6”.

Matriz de confusão com “essembly” de 6 serviços e mapeamento fonético

- Considerar todos os números não classificados “X” como 6 aumento a precisão de 91.99% para 93.41%
- Além disso, a taxa de sucesso aumentou de 80.31% para 85.15%

- Uma fraqueza identificada é o fato de poder solicitar quantos desafios quiser, até obter maior confiança na solução.

# Velocidade de resolução

- De forma serial – em média 22.24 segundos
- De forma paralela- cerca de 5.42 segundos (desvio padrão 1.25seg)
- A duração do áudio do captcha dura em média 19.13 segundos( desvio padrão de 2.39seg)
- Foi permitido inserir a solução antes mesmo de ouvir a mensagem.
- unCaptcha resolve em média com 28.3% do tempo que um humano levaria para solucionar.

# Tempo de resposta dos serviços

| Service | API Limits |
|---|---|
| Google | 60 minutes/month |
| Google Cloud | 60 minutes/month |
| Bing | 5,000 queries/month |
| IBM | 1,000 minutes/month |
| Sphinx | None |
| Wit | None |

Limites de uso para usuário Free

- Em média 2.95 segundos( desvio padrão de 1.99seg).
- Depois de ultrapassar o limite o usuário deve entrar com uma nova conta ou pagar.
- Ironicamente, a criação de contas é protegida por um reCaptcha contra registro por bots.

# Estimativa do tamanho da biblioteca

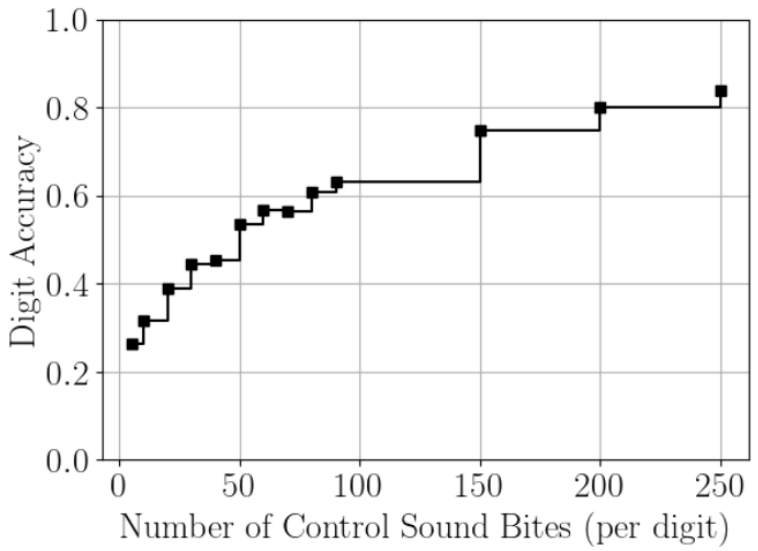
- Durante os teste, foi verificado um numero significativo de dígitos.
- Se a biblioteca de dígitos for pequena, o atacante pode baixar todos os possíveis dígitos, classifica-los e automaticamente comparar com os dígitos de um captcha não resolvido.
- Para estimar o número de duplicações foram analisados os coeficientes mel-cepstrais do sinal (MFCC).
- Foi estimado que o Google armazera aproximadamente 3600 arquivos por digito, selecionamento de forma aleatória cada um na geração do captcha
- Ao todo estima-se que a biblioteca possua 36500 dígitos.

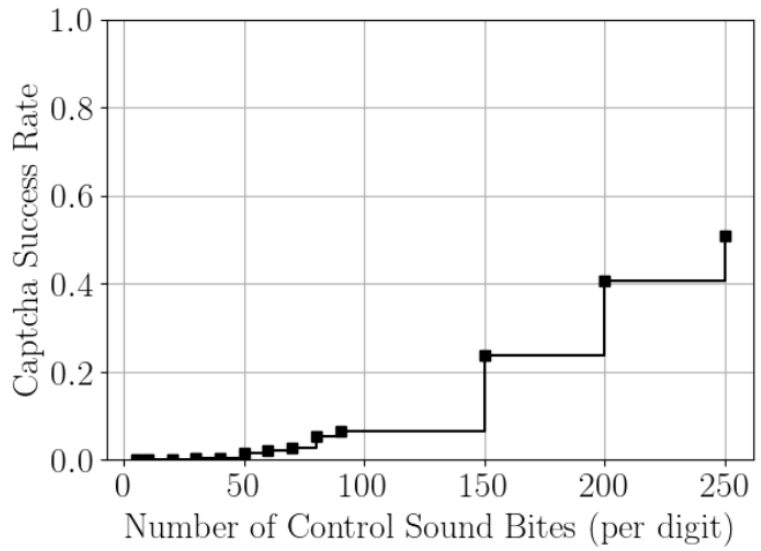
| Digit | Duplication Rate (%) | |
|---|---|---|
| 0 | 40 / 408 | (9.80%) |
| 1 | 56 / 456 | (12.28%) |
| 2 | 65 / 442 | (14.71%) |
| 3 | 49 / 463 | (10.58%) |
| 4 | 68 / 468 | (14.53%) |
| 5 | 66 / 463 | (14.25%) |
| 6 | 58 / 452 | (12.83%) |
| 7 | 67 / 466 | (14.38%) |
| 8 | 51 / 482 | (10.58%) |
| 9 | 62 / 490 | (12.65%) |
| Total | 582 / 4590 | (12.68%) |

# Ataque offline

- Sistemas de reconhecimento poderiam reconhecer os clipes de áudio do captcha e recusar-se ou transcrever de forma errada.
- O atacante poderia, antes de fazer upload, alterar o áudio com distorção ou filtro para que não fosse reconhecido como áudio de captcha
- Solução local se considerar internet limitada ou defesa desse tipo de serviço.
- Utilizou-se MFCC para comparação entre os segmentos de sons dos dígitos.
- Para reconhecer um som, os MFCCs são computador para um determinado som e para uma pequena biblioteca de "sons de exemplo".
- Utilizando Dynamic Time Warping(DTW) são comparados os coeficientes da amostra com o do banco de dados.

# Ataque offline

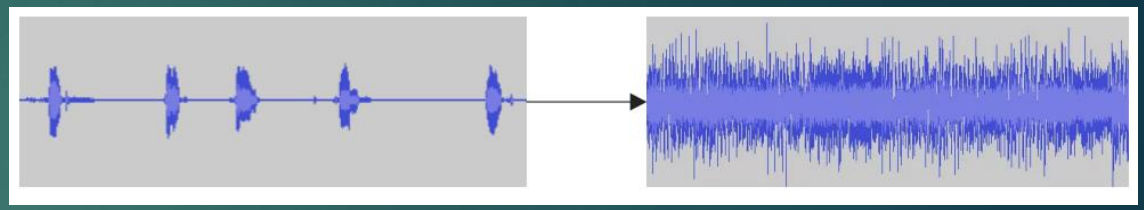


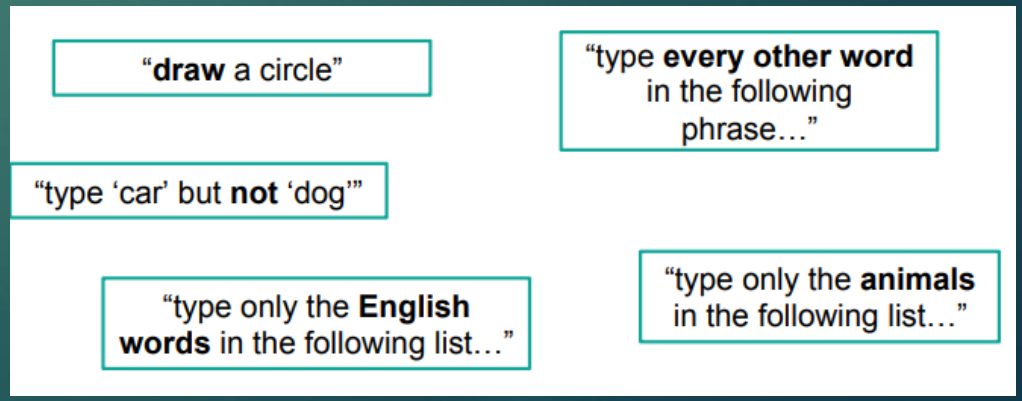




- São selecionados exemplos de controle de captchas resolvidos manualmente.
- Foram testados conjuntos de 5 a 250 exemplos de cada dígito.
- Com 250 exemplos foi alcançada uma taxa de sucesso de 51%
- O tempo de resolução é de 29.26 seg( 4 segundos mais lentos que o ataque online)

# Melhorias possíveis

- Utilizar não apenas dígitos nos áudios do captcha, mas também palavras. Apenas 10 elementos torna fácil até para algoritmos de reconhecimento off-line decifrarem.
- Adicionar distorção – pode tornar a segmentação mais difícil, porém precisa ser quieto para que não atrapalhe o reconhecimento do usuário( pode ser contornado com filtros passa-baixa )
- Utilizar frases conhecidas para que humanos decifrem pelo contexto( pode ser quebrada pelo avanço de redes neurais)
- Instruções por áudio como "digite uma palavra, "mova o mouse para cima", "escreva essa palavra e não essa outra". Adiciona uma camada ao desafio.



# Conclusão

- 450 captchas foram solucionados com 85% de precisão
- Foi relativamente fácil quebrar o áudio captcha, utilizando-se de poucos recursos.
- Isso implica que as centenas de serviços que utilizando o Google reCaptch estão vulneráveis a criação de contas por bots e ao abuso de serviço.