Rezumat

Odata cu aparitia internetului, Securitatea Sistemelor Informationale a devenit o necesitate de baza a oricarei aplicatii, iar pe parcursul anilor, implementarea acesteia a devenit din ce in ce mai puternica. De la algoritmi de criptare a datelor utilizatorilor, pana la diverse masuri de protectie anti-spam, astfel incat interfata si experienta utilizatorilor sa fie cat mai simpla, iar libertatea programatorilor, cat mai restrictionata. Una dintre aceste masuri de protectie poarta numele de CAPTCHA(Completely Automated Public Turing test to tell Computers and Humans Apart).

Exista mai multe tipuri de astfel de test, cum ar fi imagini cu diferite obiecte care trebuie identificate, imagini cu text ce trebuie transcris, si chiar si CAPTCHA-uri audio. Chiar daca rolul CAPTCHA-urilor era sa previna atacuri cibernetice si atacurile de spam asupra unui website, fiind introduse sub forme de input-form, odata cu trecerea timpului, programatorii au reusit sa creeze niste sisteme de ocolire, dezvoltand software-uri rau-intentionate, cu scopul de a rezolva CAPTCHA-ul intampinat.

In cadrul acestei lucrari, ce poarta denumirea ”Breaking CAPTCHA” ne vom axa pe tipul de CAPTCHA bazat pe text si vom observa cum putem cu ajutorul Vederii Artificiale sa construim un program care sparge CAPTCHA-urile respective. Se va explica in detaliu fluxul de lucru, obiectivul programului, algoritmii implementati, si pasii efectuati, astfel incat la final sa obtinem un soft cat mai curat si ideal. Programul va fi implementat cu ajutorul limbajului de programare Python, iar pe langa bibliotecile secundare vor fi utilizate 3 biblioteci principale dezvoltate special pentru lucrul cu Inteligenta Artificiala: NumPy, PyTorch, OpenCV. Seturile de date cu imagini CAPTCHA, descrise in lucrare, vor fi generate din librarii implementate in diverse limbaje, cum ar fi JavaScript, Python, Rust, etc.

Spre final vom incerca sa imitam un scenariu din viata de zi cu zi, si sa punem in practica bot-ul creat, utilizand o aplicatie secundara cu scopul de a observa comportamentul programului,

remarca progresele, si de a formula concluzii asupra masurii de protectie, revenind cu o potentiala extensie a studiului efectuat, posibile imbunatatiri ale sistemului anti-CAPTCHA creat, dar si cu o sinteza precum si o serie de recomandari in care vor fi exemplificate tehnicile de protectie care opun o rezistenta mai mare, unor astfel de tipuri de atac.

Cuprins

Introducere

Odata cu dezvoltarea internetului si cresterea numarului de persoane care au acces la platformele web, a fost necesara introducerea unor sisteme care impiedica sau provoaca inconveniente in automatizarea proceselor, cum ar fi creare excesiva de conturi, colectare de date, reclama fortata, cu alte cuvinte, sisteme care implementeaza masuri de protectie impotriva botilor. Astfel a fost introdusa CAPTCHA.

CAPTCHA este prescurtare de la Completely Automated Public Turing Test to Tell Computers and Humans Apart.

Ce este Testul Turing? Testul turing original denumit si ”jocul de imitatie”, creat de Alan Turing, este un experiment implementat pentru a testa capacitatea masinii de a manifesta un comportament inteligent, echivalent, sau chiar si indisctintibil fata de cel al oamenilor. [1]

Termenul de CAPTCHA a fost inventat in anul 2003 de catre Luis von Ahn, Manuel Blum, Nicholas J.Hopper, si John Langford, iar primul si cel mai comun tip de CAPTCHA a fost inventat prima data in 1997 de catre 2 grupe de oameni de stiinta lucrand in paralel [2]. Mai jos se poate observa prima versiune a CAPTCHA-ului.

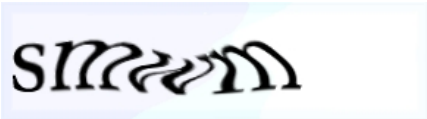

Figure 1: First Version of CAPTCHA

Dupa cum se poate observa, prima versiune de CAPTCHA este una bazata pe text, dar pe langa aceasta pe parcusul evolutiei au fost dezvoltate si alte tipuri cum ar fi, bazate pe imagini, sau audio. Initial s-a constatat ca CAPTCHA-urile puteau fi folosite in scopuri de securitate deoarece nu se stia, nu era descoperit, sau inventat un program care ar reusi sa rezolve problema

intampinata

O vulnerabilitate majora, care nu era considerata la momentul crearii CAPTCHA-urilor datorita nepopularitatii si progreselor minore, era si ramane un alt subdomeniu al informaticii, Inteligenta Artificiala, si anume Vederea Artificiala (Computer Vision), tinta caruia este sa imite comportamentul uman, sa detecteze, si sa identifice obiecte de interes intr-o imagine, secventa video, etc.

In aceasta lucrare ne vom axa pe CAPTCHA-uri de tip text, de altfel cum am si mentionat mai sus, cel mai comun tip de CAPTCHA, vrem sa evaluam starea curenta a tehnologiilor AI experimentand cu diverse dataseturi de CAPTCHA bazate pe text si sa determinam daca inca mai este valida ipoteza creata, si mai exact, daca CAPTCHA-urile de tip text manifesta o rezistenta puternica impotriva programelor automatizate, boti.

Aplicatia principala este un script de python care ruleaza pe fundal face http-request catre aplicatia secundara, care e o platforma web, si creeaza in continuu conturi pentru platforma respectiva, folosindu-se de AI-ul antrenat pentru rezolvarea CAPTCHA-ului corespunzator

Aplicatia secundara este o aplicatie web, vulnerabila, simpla, creata cu ajutorul framework-urilor Nuxt.js pentru FrontEnd si .Net Framework pentru BackEnd, care serveste ca scop, ilustrarea functionalitatii aplicatiei principale

Structura lucrarii este urmatoarea:

Capitolul I: Prezentarea generala a CAPTCHA-urilor. Vor fi prezentate librariile de Captcha utilizate pentru generarea seturilor de date, si folosite pe parcursul experimentelor, vor fi ilustrate tehnicile de protectie care sunt folosite, evolutia generala a lor, si un tabel cu situatia actuala.

Capitolul II: Prezentare generala a Invatarii Automate Adanci. Vor fi descrise bibliotecile folosite, vor fi explicati algoritmii si modalitatile de combatere a tehnicilor de protectie intampinate.

Capitolul III: Experimentele realizate si rezultatele respective. Alegerea generatoarelor de imagini CAPTCHA tinta, urmata de justificarea modalitatii de spargere, proiectarea sistemu-

lui AI si compararea acestuia cu SOTA existent.

Capitolul IV: Vor fi structurate sub forma de sinteza concluziile la care am ajuns, problemele pe care le-am intampinat, niste propuneri cu ce se poate imbunatati, cat si recomandari personale in privinta alegerii masurilor de protectie.

In continuare vom vorbi doar despre CAPTCHA bazat pe text, si pentru simplicitate ii vom spune doar CAPTCHA

Capitolul 1

Istoric CAPTCHA

Primul CAPTCHA (Figura 1) a fost folosit in 1997 de catre motorul de cautare AltaVista si a impiedicat boti sa adauge URL-uri (Uniform Resource Locator) la motorul lor web [3]. Putem observa masurile de protectie intreprinse in cazul respectiv, caracterele sunt distorsionate, iar imaginea de fundal are cromatica unui gradient. In trecut doar aceste lucruri erau de ajuns pentru asigurarea protectiei, impotriva programelor rau intentionate, dar datorita progresului in stiinta, bariera a fost ocolita de catre un soft OCR (Optical character recognition) implementat in anii 2000 [3].

CAPTCHA-urile au devenit mai complexe, diverse nivele de zgomot si distorsii au fost adaugate la imagini, astfel incat OCR-urile populare au devenit ineficiente iar taskul pentru spargerea CAPTCHA-urilor a devenit costisitor pentru partea atacanta. Developerii de CAPTCHA in schimb trebuiau sa aiba grija cu nivelul de anti OCR aplicat imaginilor intrucat uneori se producea o inconvenienta mai mare pentru om decat pentru calculator sa le recunoasca [3].

O versiune mai complexa a primului CAPTCHA a fost implementata in Rusia, pentru platforma web Yandex.com.

Figure 2: Improved CAPTCHA

Remarcam un nivel de distorsie mai mare a caracterelor in imagine precum si diferentierea in culori a textului combinat cu imaginea de fundal ce corespunde unei valori aleatoare intr-un

spectru (0 - 255) de la negru la alb. Deasemenea imaginea de fundal este inconsistenta, tehnica care eroneaza si previne atacuri de spargere care utilizeaza anumite euristici precum diferenta neschimbata intre culorile imaginii, aceasta fiind cea mai mare problema a atacatorului - separarea caracterelor de imaginea de fundal. Astfel pentru a avansa in spargerea CAPTCHA-ului este necesara identificarea fonului si setarea fiecarui pixel al sau cu 0 (negru) iar fiecare pixel care apartine unui caracter cu 1 (alb) procedeu care se numeste binarizarea imaginii [3].

In 2005 isi face debutul primul CAPTCHA creat de Google, cu numele reCAPTCHA.

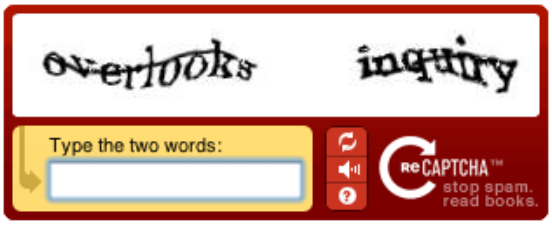

Figure 3: Google reCAPTCHA

Ca masuri de securitate, putem remarca multiple cuvinte, forma de valuri a acestora, precum si o linie peste caractere care impiedica segmentarea usoara a lor. reCAPTCHA-ul se deosebeste prin modul in care a fost programat, cuvintele fiind scanate dintr-o carte aleasa de adminului programului, si alese prin ajutorul Crowd-Funding-ului.

Programul reCAPTCHA alegea cuvintele in felul urmator, unul care e lizibil de catre OCR si celalalt nu. Apoi CAPTCHA-urile erau pasate unui grup de utilizatori, care pe rand rezolvau testele respective. Daca primul cuvant era introdus corect, programul presupunea ca si al doilea cuvant este corect, al doilea cuvant fiind pasat urmatorilor utilizatori ca fiind primul in testele noi. Programul mai apoi compara raspunsurile si avea destule dovezi pentru a recunoaste care cuvinte au fost introduse corect.

Astfel programul avea un scop dublu:

- Verificarea utilizatorului - Om/Robot?
- Verificarea cuvantului care nu putea fi citit de catre OCR si digitalizarea acestuia pentru cunoasterea umana. [4]

Ideea Google-ului era sa digitalizeze un numar cat mai mare de carti, ajungand la un numar de 40 milioane in 2019, potrivit wikipedia[5], si de a determina oamenii sa identifice caractere si cuvinte care nu puteau fi identificate de cei mai buni algoritmi de procesare a imaginilor existente la acel moment [4]

In acelasi timp multi programatori construiau diverse tipuri de CAPTCHA implementand multiple tehnici de protectie, care pot fi urmarite in imaginea de mai jos:

Figure 4: Tehnici de Protectie Captcha

Tehnici de Protectie

- Distorsiunea Caracterelor - Orice fel de modificare a caracterului cum ar fi rotirea sau deformarea acestora impiedica botii inspre utilizarea potrivii sabloanelor, cu niste imagini ale caracterelor prestabilite, pentru identificarea rapida a caracterelor distorsionate.

- Distorsiunea Imaginii de Fundal - Aplicarea diferitor manipulari asupra fonului, cum ar fi diverse culori, sau imagini inconsistente, acopera o mare vulnerabilitate in spargerea CAPTCHA-ului, problema de baza, si cea mai complicata fiind separarea fonului de caractere.

- Introducerea zgomotului in imagine - Zgomotul in imagine aplicat corect reuseste sa uneasca caracterele si imaginea de fundal astfel acestea raman lizibile pentru om dar foarte greu de distins de catre calculator, iar incercarea eliminarii acestuia, duce catre eliminarea unor componente foarte importante in CAPTCHA, cum ar fi portiuni de caractere sau chiar caractere intregi.

- Utilizarea diferitor fonturi - Stilul, marimea si originea fontului este o aplicatie importanta, fiindca adauga complexitate si provoaca inconveniente botilor in replicarea si identificarea caracterelor .

- Amplasarea aleatoare a caracterelor - Pozitia caracterelor in CAPTCHA, pot cauza erori in programul malitios si euristica utilizata de catre atacator.

- Concatenarea caracterelor - Atacatorul trebuie sa depuna efort inspre proiectarea unui program care separa caracterele concatenate, astfel incat imaginea este segmentata corect.

- Adaugarea elementelor secundare - Elemente secundare cum ar fi puncte, linii, figuri geometrice aleatoare in imagine sunt si ele considerate fiind zgomot, dar e mai putin aleator,

si reprezinta o piedica in segmentarea imaginii, detectia precum si recunoasterea caracterelor.

- Lungimea textului - O tehnica simpla dar puternica, se bazeaza pe faptul ca OCR-ul nu va reusi sa identifice corect toate caracterele din CAPTCHA, prin urmare daca lungimea textului este 15, iar OCR-ul a ghicit doar 14 caractere, testul va fi considerat incorect

- Combinatii intre tehnici - Evident combinarea a doua sau mai multe procedee descrise mai sus, sporeste complexitatea testului prin urmare si dificultatea intampinata de algoritmii automatizati este mai mare

In continuare atasez un tabel cu generatoarele de CAPTCHA care au fost analizate si in cadrul carora au fost executate experimentele din lucrarea respectiva. Librariile prezentate au fost construite de alti programatori, si se pot gasi open-source, aferent link-urilor atasate acestora.

table goes here

References

[1] https://en.wikipedia.org/wiki/Turing_test

[2] https://en.wikipedia.org/wiki/CAPTCHA

[3] https://habr.com/en/articles/670520/

[4] History and Evolution of CAPTCHA, Kishan Pandey

[5] https://en.wikipedia.org/wiki/Google_Books